WIRELESS PANT AND TILT SURVEILLANCE PLATFORM

By

EVI VIDYANA RIAZUDDIN

FINAL YEAR PROJECT
FINAL REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

WIRELESS PAN AND TILT SURVEILLANCE PLATFORM

by

Evi Vidyana Riazuddin

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

____

Mr. Patrick Sebastian
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

December 2007

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Evi Vidyana Riazuddin

ABSTRACT

The main objective of this project is to have a wireless pan and tilt platform. One platform will track an object or person of interest until it is out of the camera viewfield or platform limit. The next camera on the second platform will continue tracking the image. This project involves the study on wireless system available and how these wireless systems can be implemented into the project. The examples of wireless system available are Bluetooth, wifi, Radio frequency and infra red. Research has been made into the availability of each wireless system and the task is focusing on the transferring the input data wirelessly from computer to the servo motor by using RF and also bluetooth module. Testing method is developed by doing some experiment in transferring the data and the performance measurements is taking into consideration in completing the project. For the RF module, C codes for PIC microcontroller at the transmitter and receiver side is developed. For the Bluetooth module, the device is setup so that to transfer data wirelessly once the link is established.

ACKNOWLEDGEMENTS

Firstly, I would like to thank our Almighty God for giving me the strength and time to complete this Progress Report.

I would like to grant my most gratitude to my Supervisor, Mr. Patrick Sebastian for supervising my Final Year Project and giving me endless guidance and advice on the work that we should do in order to complete the project successfully. He has never failed to give me support and was always understanding despite all the errors and delays that I made during the completion of the project.

Apart from that, to the technician Mr. Musa bin Mohd Yusuf, for the support and most importantly for the cooperation and help that had been given. Also special thanks to my fellow colleagues who have aided me with valuable advises and opinions regarding the project. Last but not least, to my beloved parents and family for their continuous love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

A/D	Analog to digital
AP	Access Point
ASK	Amplitude Shift Keyed
CW	Clockwise
CCW	Counter Clockwise
DC	Direct Current
I/O	Input Output
LAN	Local Area Network
MATLAB	Matrix Laboratory
PC	Personal Computer
PIC	Programmable Interface Controller
PWM	Pulse-width Modulation
RC	Remote Control
RF	Radio Frequency
RISC	Reduced Instruction Set Computer
SNR	Signal-to-Noise Ratio
TE	Transmit Enable
TTL	Transistor-Transistor Logic
USART	Universal Synchronous/Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VT	Valid Transmission

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Wireless connection application was developed to reduce dependency on wired devices. To make life become more sophisticated and simple, this is when a wireless type of video comes into consideration. As for video conferencing devices, a wireless camera is completely mobile units that deliver images direct to PC, laptop or dedicated monitoring station via wireless application. The camera should recognize the suspects based on database of suspect figures. Once the system recognizes the suspect, the camera should follow the movement of the suspect.

1.2 Problem Statement

The current pan and tilt surveillance platforms are wired connected device, the application of the device is limited. Once the device is installed, it cannot be move to cater for a better view. As for the current system, the separate data incoming from webcam is transferred via universal serial bus (USB) port while the movement control of the platform will be done by MATLAB, which will then sending the signal for the motor movement via serial port. Thus, for outdoor and wide installation the application becomes difficult. By developing the wireless technology into the project, one can place the camera everywhere without worrying about the tedious installation.

1.3 Objective

- To enhance the design of the existing pan and tilt surveillance platform by make it wireless.

1.4 Scope of Study

The scope of this project includes the study of wireless communication system using RF module, Bluetooth or others wireless system available in transferring the desired signal from PC to platform motor wirelessly. The main concern is to receive the correct data through serial communication port which consist information to move the camera. The project also includes the construction of the circuit design for data transmission. It is also involves the development of tracking algorithm for real time video surveillance utilization.

CHAPTER 2
THEORY

2.1 Bluetooth

Bluetooth is an open standard specification for radio frequency (RF) based short-range connectivity technology that promises to change the face of computing and wireless communication [1]. Its main strength is its ability to simultaneously handle both data transmission and voice transmission. It's capable of supporting one asynchronous data channel and up to three synchronous voice channels [1]. Bluetooth offer the facility for collaboration between devices, in the proximity of one another. The aim is to eliminate the usage of cable. This system operates in worldwide unlicensed 2.4GHz Industrial Scientific Medical frequency band. At least 2 nodes are required to form Bluetooth network, either one of the nodes can be master. Master will search the connection once the link is established.

2.1.1 KC-21 Bluetooth Module

This Bluetooth receiver module support TTL level UART communications. The minimum UART pins that are needed are the Rx and Tx pins. When the device is ON, the module will be connected to the PC via a bluetooth dongle. The device is then select to connect to serial port service by Interface using HyperTerminal. This module is surface mounted PCB which includes 14 general purpose input/output lines. This module provide fully embedded, ready to use wireless technology. The datasheet for bluetooth this module is attached in APPENDIX A.

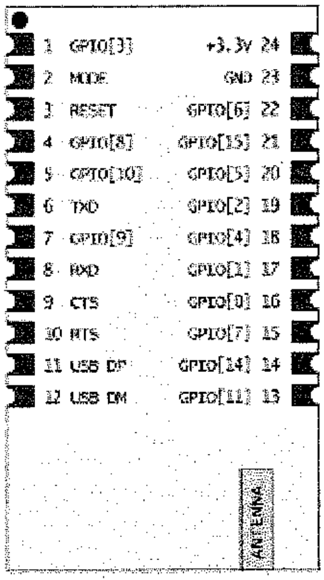


Figure 1 Bluetooth KC-21 Module [8]

Table 1 KC-21 Pin Configuration [8]

Name	**Type**	**Pin Number**	**Description**
RxD	I	8	Received Data
TxD	O	6	Transmit Data
CTS#	I	9	Clear to send (Active Low)
RTS#	O	10	Request to Send (Active Low)
DM	I/O	11	USB Data minus
DP	I/O	12	USB Data plus
MODE	I	2	Reserved
V_{DD}		24	V_{DD}
GND		23	GND
RESET#	I	3	Reset Input (Active Low for 5 ms)
GPIO [0-15]	I/O		General Purpose Input/Output

2.2 RF Transmission

A radio wave is an electromagnetic wave propagated by an antenna. Radio waves have different frequency, and by tuning a radio receiver to a specific frequency one can pick up a specific signal. Many wireless technologies are based on RF field propagation. When an RF current is supplied to an antenna, an electromagnetic field is created that then is able to propagate through space. To provide the RF transmission, RF transmitter and receiver modules used to allow the implementation of data link at speed. As for this project, the transmission process will utilize the Amplitude-shift keying (ASK) method. ASK is a form of modulation that represents digital data as various in the amplitude of a carrier wave [2]. As shown in Figure 2, the carrier wave is multiplied by the digital signal $f(t)$. Thus, modulated carrier signal transmitted $S(t)$ is described is the mathematical form below.

$$S(t) = f(t)sin(2\pi f_c t + \theta)$$

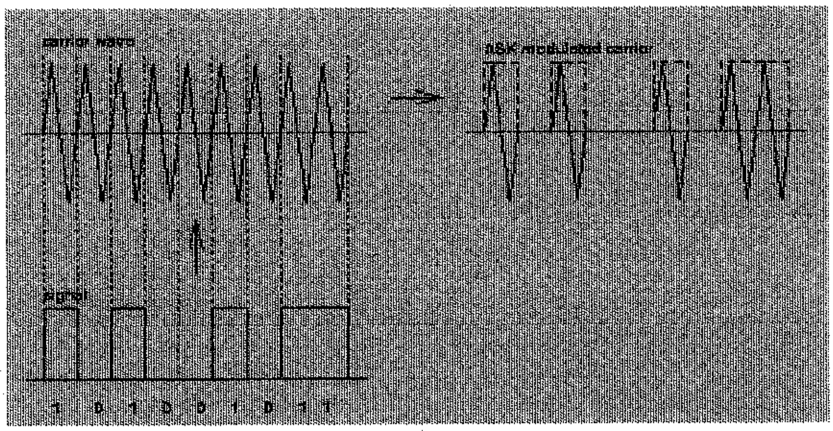


Figure 2   Amplitude-shift keying [3]

2.2.1 TLP-434A Transmitter

The RF transmitter module used for this circuit is TLP-434A. This unit operates from 2 – 12VDC. A range up to 200m is possible. The transmitter will modulate the data which received from the binary 12 bit encoder, HT-12E by utilizing the amplitude shift keying, ASK, method. The frequency for the transmission is 433.92MHz. Datasheet for the transmitter is attached in the APPENDIX B. Using the Digital Data Input (pin 6) command, the content of the data are transmitted.

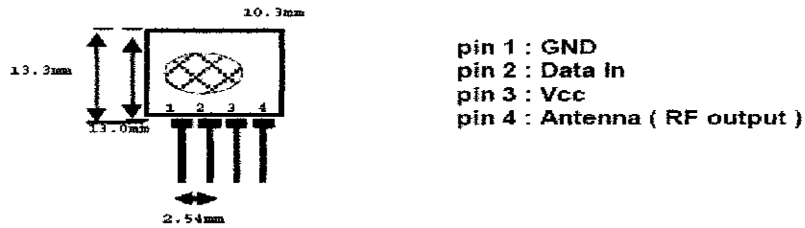


Figure 3 TLP-434A RF Transmitter [9]

2.2.2 *RLP434A Receiver*

Data is gathering by using receiver PLP434A. As shown in figure below, this receiver module consists of 8 pin. Pin 2 digital output data will be used for transferring the data to decoder. There are several important parameters such as signal-to-noise ratio, data rate and bandwidth that need to be considered in determining the successful a receiver interpreting an incoming signal. An increase in data rate, increases bit error rate. An increase in SNR, decreases bit error rate and an increase in bandwidth allows an increase in data rate. Datasheet for this Receiver is attached in APPENDIX B.

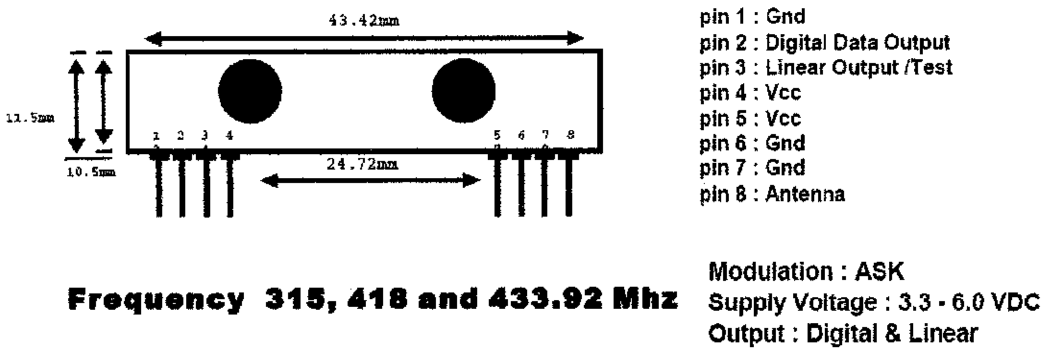


Figure 4 TLP-434A Receiver [9]

2.2.3 *Encoder*

Before transmitting the data to the transmitter, HT12E encoder is used to encode the data obtained from the computer. A data stream will be encoded onto a carrier signal that will than be propagated by means of an electromagnetic wave which called modulation process. Encoding technique is used so that the circuit construction is less complex. The data will than can are transmitted together with the header bits via the RF transmitter

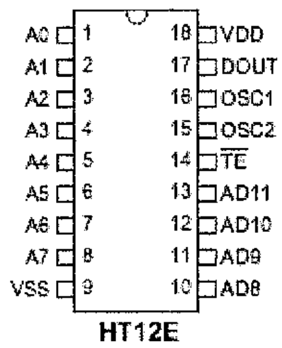


Figure 5 Encoder 18 pin [5]

The series of encoders begin with 4 word transmission cycle when the transmission is enabled. Once the trigger signal, TE, is receiver, the RF transmission medium will then transmit the date together with the header bit. Once the transmission disable, the encoder will stop sending the data. Encoded data serial transmission output will be sent to the transmitter via pin 17. Datasheet for this encoder is attached in APPENDIX C.

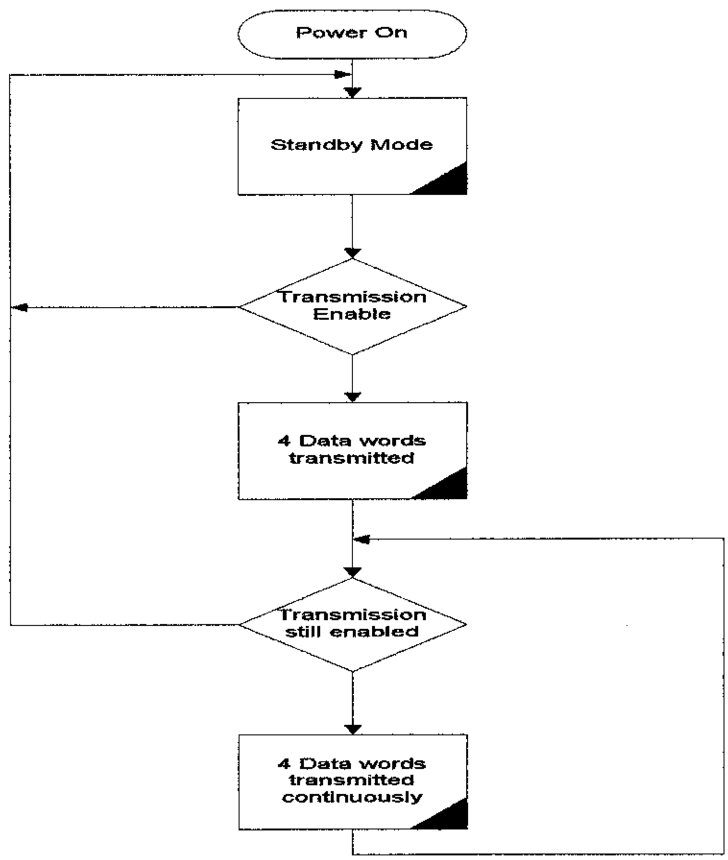


Figure 6 Flow chart of the encoder [5]

As shown in Figure 6, once the encoder is enabled by giving a HIGH input at pin 14, the 4 bit data will be transmitted. Transmission process will be continued as long as the TE pin is in the enable.

2.2.4 Decoder

HT12D decoder will receive serial address and data from a programmed series of encoder that are transmitter via RF transmitter. This decoder will compare the serial input data a few times with their local address. The data will be successfully decoded and transferred to the output pin if it is matched with the local address. There are VT pin to indicate a valid transmission. Detains description regarding this encoder is attached in APPENDIX D.

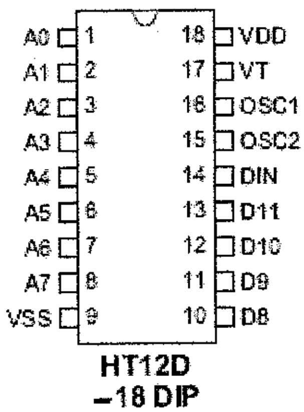


Figure 7 Decoder 18 pin [6]

2.3 MAX232

MAX232 is used to convert RS-232 level down to lower level which is 5 volt. The RS-232 signals are represented by voltage levels with respect to a system common power or logic ground and its work with voltages -15V to +15V for high and low. On the other hand, TTL logic operates between 0V and +5V. Thus, the RS-232 signal level is far too high and has to be reduced. Figure 3 below shows the schematic for connection between RS-232 to MAX 232. Pin 14 and 13 will be connected to PC DB9 Female. The datasheet for MAX232 is showed in APPENDIX E.

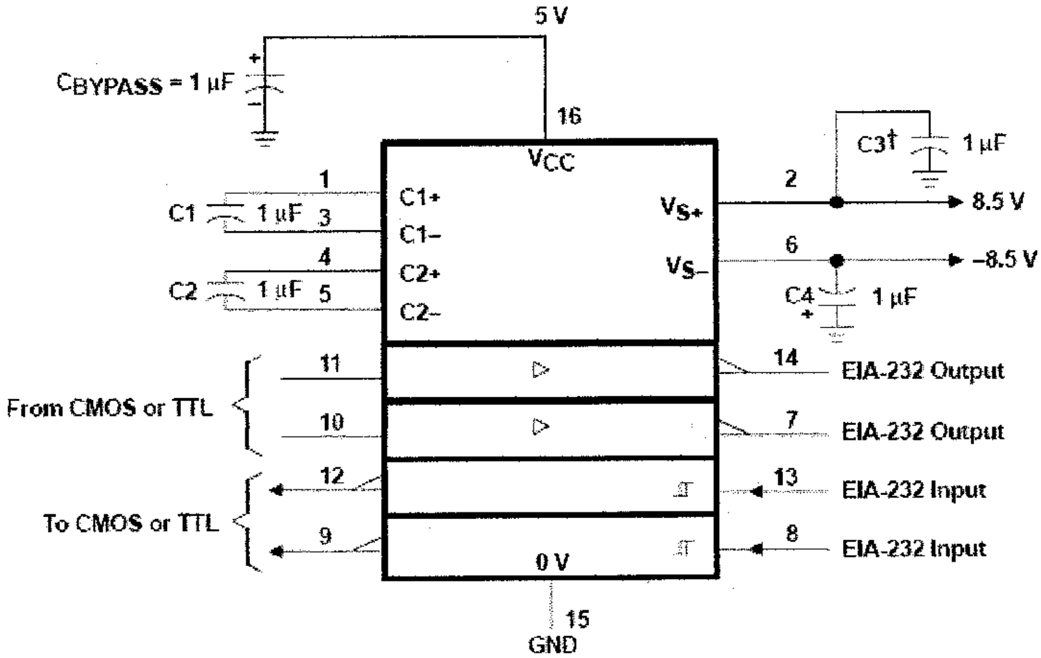


Figure 8 MAX232 circuit [10]

2.4 Serial Communication Interface RS-232

The communications links across which computers or parts of computers talk to one another may be either serial or parallel. Parallel link transmits several streams of data along multiple channels, where all the bits of each symbol are sent together. Whereby, serial link transmits a single stream of data. The data are sent one bit at a time, sequentially, over a communications channel or computer bus. In this project, serial is a better option because it is cheaper and easier to be implemented. Many ICs have serial interfaces, as opposed to parallel ones, so that they have fewer pins. This serial port will transmit a '1' as -3 to -25 volts and a '0' as +3 to +25 volts where as a parallel port transmits a '0' as 0V and a '1' as 5V. Therefore, serial communication can cater for higher range of voltage level compared to parallel communication. For this project 9 pin female connectors is used. The pin configuration for the connector is listed in Table 2 below. Only two pins are commonly used to communicate with the microcontroller which is the Transmit Data (TXD) and Receive Data (RXD) pins.

Table 2 DB9 Female connector

Pin	Signal
1	Data carrier detect
2	Received data
3	Transmitted data
4	Data terminal ready
5	Signal ground
6	Data set ready
7	Request to send
8	Clear to send
9	Ring Indicator

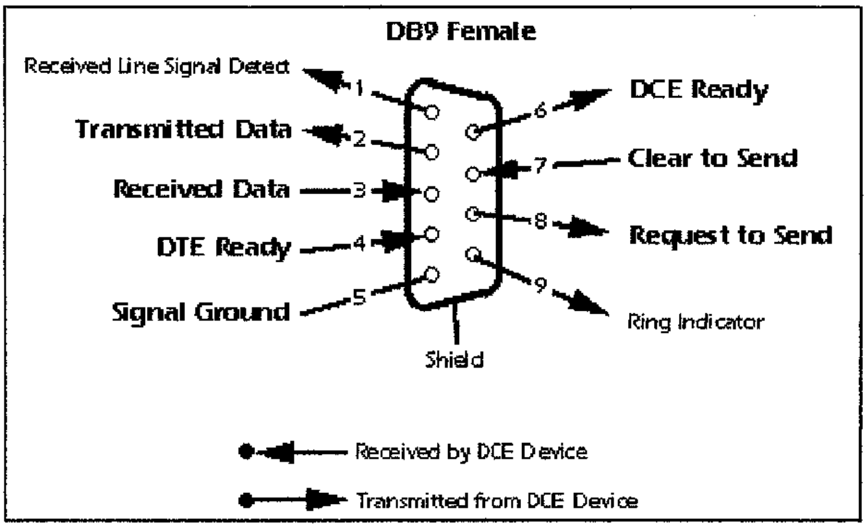


Figure 9 Pin configuration for DB9 female connector

2.5 PIC16F84A Microcontroller

The PIC16F84A employs an advance Reduced Instruction Set Computer (RISC) single-cycle microcontroller equipped with 13 input/output pin [4]. This microcontroller had enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with a separate 8-bit wide data bus. The two stage instruction pipeline allows all instruction to execute in a single cycle except for program branches which is required two cycles. A total 35 instruction are available with an additions of large register set is used to achieve a very high performance level. Datasheet for this Microcontroller is attached in APPENDIX F.

2.6 PIC16F877 Microcontroller

There is the option of using one microcontroller at both transmitter and receiver side. The data from the receiver is read by the microcontroller and than transmitted. At the receiver side, it is read into the flash of the receiving microcontroller. The microcontroller will be needed to set the address bit of the encoder and the transmission bit. For this project, The PIC16F877 is used at the receiver side. The PIC16F877 has 5 digital I/O ports (A-E) each between 3 and 8 bits wide. Each port is mapped into the register space, and may be read and written to like any other register. The circuitry is such that it is not possible to physically input to and output from a particular pin simultaneously. For most ports, the I/O pins direction (input or output) is controlled by the data direction register, called the TRIS register. TRIS<x> controls the direction of PORT<x>. A `1' in the TRIS bit corresponds to that pin being an input, while a `0' corresponds to that pin being an output [7]. The PORT register is the latch for the data to be output. When the PORT is read, the device reads the levels present on the I/O pins (not the latch). This means that care should be taken with read-modify-write commands on the ports and changing the direction of a pin from an input to an output. The pins on the PIC16F877 are multiplexed so that one of several functions may be selected (e.g. pin 2 may be used as either bit 0 of I/O port A, or as channel 0 of the A/D converter). The TRIS registers control the direction of the port pins, even when they are being used as analog inputs [7]. The user must ensure the TRIS bits are maintained set when using the pins as analog inputs. The A/D allows conversion of an analog input signal to a corresponding 8-bit digital number. The output of the analog sample and hold is the input into the converter, which generates the result via successive approximation. The analog reference voltage is software selectable to either the positive supply voltage (5V) [7]. Datasheet for this Microcontroller is attached in APPENDIX G.

2.7 Servo Motor

For this project, standard type of servo motor is used to move the camera. Command from the MATLAB will be executed and the motor will move the camera to the desired location. Servo motor are comprised of a DC motor mechanically link to a potentiometer. This motor is moved base on Pulse-width modulation (PWM). 1 millisecond pulse train will position the servo motor to the left. 2 millisecond pulse train positions the servo motor to the right and 1.5 millisecond pulse train will position motor at the 90° position.

Figure 10 Servo motor

CHAPTER 3
PROJECT WORK

In completing the project, all required procedures have been identified and as illustrated in the Figure 11 below. All procedures are carried out subsequently at all time. The procedure is continuously update as the project go so that to cater for the need in completing the project.

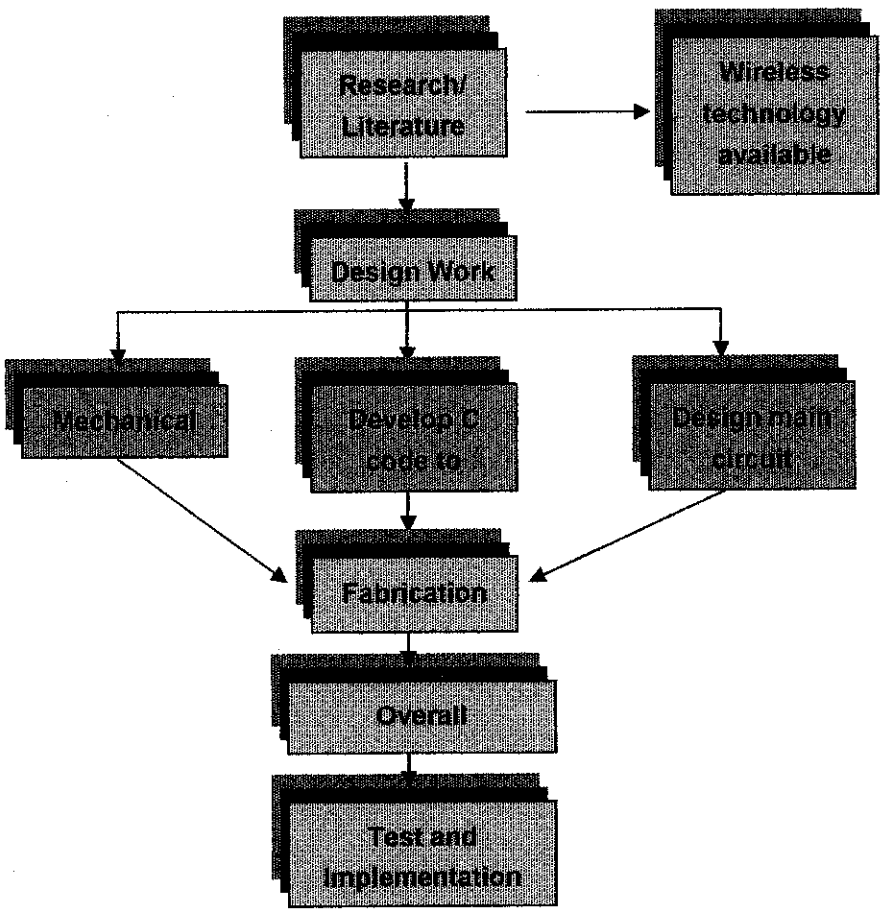


Figure 11 Procedure identification in completing the project

3.1 Literature review and analysis

Research is the prerequisite in the early stage of the project as to pursue on the next procedures of the project. It is also to develop testing methods, performance measurement and also to collect result from other works. Apart from that, comparative study can be done during this stage and conclusion can be made base on the analysis during the research. The examples of the material used are encyclopedias, technical papers, datasheets, conference paper and journals. A few solution methods in implementing the wireless part for project are obtained so that comparison analysis can be made before embarking on the selected solution. For project, the work done is focusing on using RF and Bluetooth module in data transmission.

3.2 Circuit design and modeling

Circuit designed is carried out once the literature review is completed. The circuit designed is base on the desired system implemented in the project which is RF and Bluetooth module implementation. Datasheet and technical paper is referred in making sure the correctness of the circuit wiring. Once the circuit is constructed, a few testing is developed so that analysis can be made base on the result. Modification of the circuit is made along the design process to get the better outcomes. The chosen RF module solution to provide the wireless transmission consists of Transmitter and receiver from Laipac Technology is used. HT-12E and HT-12D type of encoder and decoder is recommended to be used with this RF module. This type of implementation requires the interfacing with RS-232 and also the need of microcontroller to program the data to obtain the desired output. A block diagram of the RF solution is shown in figure 12.

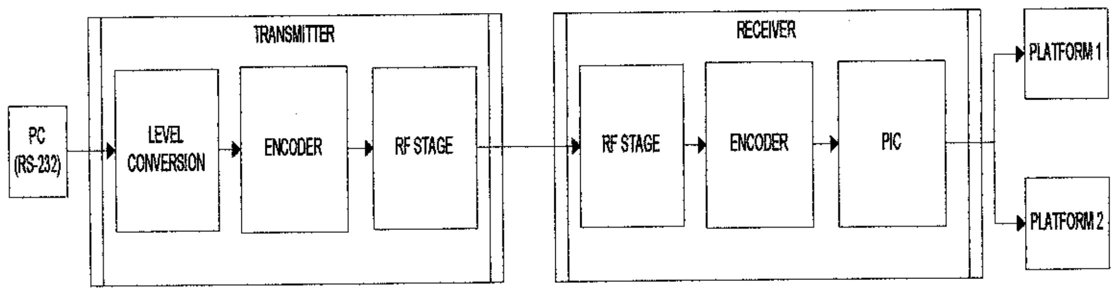


Figure 12 Block diagram for the transmitter and receiver

For the bluetooth solution, computer is used to be the host. After the computer is configured by using bluetooth dongle, the bluetooth module is automatically accept connection request and set to connect true bluetooth serial port. Connection is done using HyperTerminal. A block diagram of the RF solution is shown in figure 13.

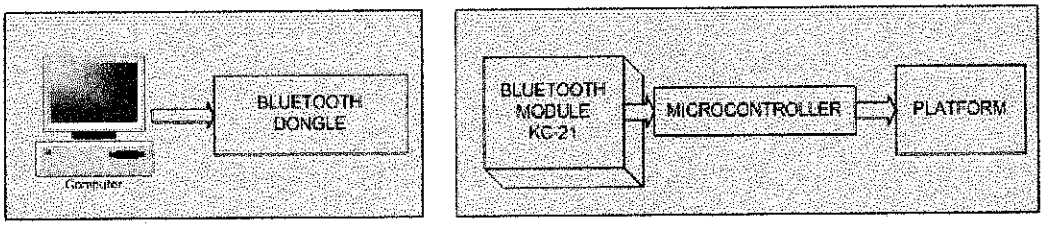


Figure 13 Block Diagram for bluetooth connection

Self study is made during this stage so that to further understand the desired data to be transmitted. Streams of bits that represent character values form PC is sent to the microcontroller for data transmission. Apart from that, knowledge in C programming and data communication network is required in completing this task.

3.3 Integrating software design with hardware

Platforms for web cam mounting are being built. To make the servo motor move in desired position, experiment has been conducted. Understanding on how to move the servo motor by using microcontroller is needed. It utilizes the knowledge in C programming and the technical knowledge to build the circuit.

3.4 Tools required

This project required integration of software and hardware in order to complete the entire project. The following list shows the hardware and software used throughout the completion of the project

3.4.1 Hardware

- Drilling equipment
- RF Module
- Bluetooth Module

- Bluetooth dongle
- Personal Computer
- Breadboards
- Resistor/Capacitor/LED/Switch
- 4MHz and 20MHz crystal oscillator
- RS232
- MAX232
- Microcontroller 16F877/16F84
- Servo Motor
- Stainless steel plate
- Webcam and Platform

3.4.2 Software

- EAGLE Layout Editor 4.13
- PIC C Compiler
- HyperTerminal
- WARP13
- IVP BlueSoleil

CHAPTER 4
RESULT

4.1 Platform construction

For this project, two unit of platform is required. 160 x 160 x 85 mm size of plastic boxes is used to be the base for the platform. Two units of servo motor are attached to the platform to perform the pan and tilt operation. The servo motor is installed by attaching to a 1 mm thickness of galvanized-plate.

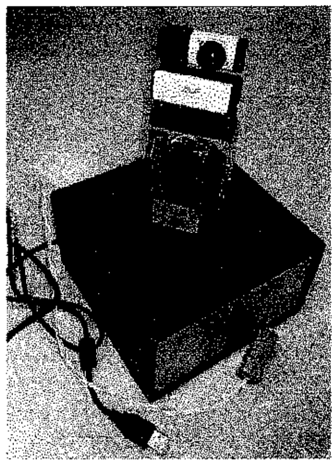


Figure 14 Camera platform

4.2 Servo Motor Movement

Servos have a very simple electrical interface; they usually have 3 wires, one for power, one for ground and the other for the pulse train. Once power (4.8V - 6.0V) and ground is supplied to the servo, the data wired is prepared to receive encoded signal in the form of pulse width modulation (PWM). The duty-cycle is of 20ms, and the pulse width contained within those 20ms varies from 1ms to 2ms. It is these variations that control the servo. If a 1ms pulse is constantly fed to the servo, it will position itself around -60° angle, 1.5ms and it will go to the center 0°, and 2ms will position the servo at 60° angle Circuit is constructed so that to have a basic understanding on how to move the motor by decision based on predetermined situation and selection input which programmed in the microcontroller. The schematic for the servo motor circuit is attached in the APPENDIX H. Table 3 below shows the angle movement base on input given to the microcontroller

A0	A1	A2	Angle Movement
0	0	0	No movement
0	0	1	90° clock wise
0	1	0	45° clock wise
0	1	1	90° counter clock wise
1	0	0	45° counter clock wise

Table 3 Input pin to the PIC to move the servo motor

The selected input at pin A0, A1 and A3 will then activate the servo motor at pin B0. C Programming to control the servo motor is attached in APPENDIX M.

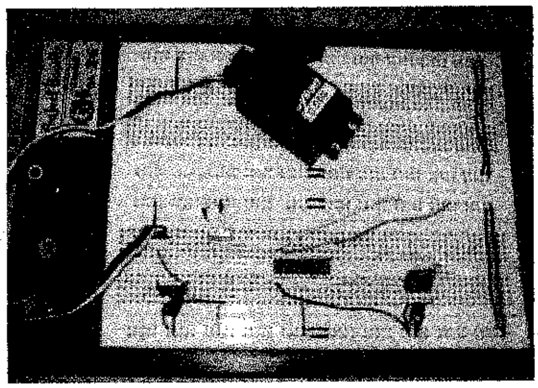


Figure 15 Servo motor circuit

4.3 Data Transmission for PC to PIC 16F877 Microcontroller

This procedure is used to test how to handles input from the RS-232 serial communication and transfers the input to the Microcontroller by looking at the incoming and outgoing data. As for this project, the PIC16F877 which has an integrated USART used to communicate to the devices such as hyper terminal and the serial port monitor tool in CCS-PICC. These monitoring devices are used for testing and troubleshooting so that to make sure there are data transmission. These devices will transmit serially the characters typed on the keyboard. From the serial port, a built in function is used to receive the character from the keyboard. For example, if 'w' is typed, the ASCI code for this character is 0x77. The output can be displayed on the PORT D by using the LED. The hex value is first converted to decimal which equal to 01110111. Thus, LED on PORT D0, D1, D2, D4, D5 and D6 will be on. For this project, the program is modified so that to move the servo motor by typing input from the computer. C code is attached in the APPENDIX N and the circuit diagram for this part is attached in APPENDIX I.

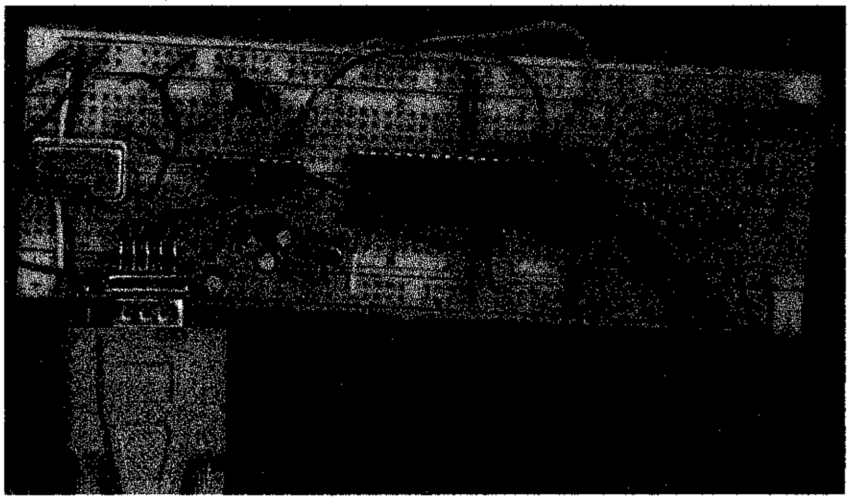


Figure 16 Circuit for connecting PC to Microcontroller

4.4 Transmitter/Receiver

The first alternative solution to transmit the data wirelessly is by using a transmitter and receiver produce by the company name Laipac. The TLP and RLP 434A are simple transmitters and receiver that transmit using radio frequency at 315 MHz, 418MHz and 433.92 MHz. to transmit the data, an encoder and decoder would be needed. HT12E encoder and HT12D decoder are currently the proposed by the Laipac to use with their product. Apart from that, there is the option of using one microcontroller at both transmitter and receiver side. The data from the receiver is read by the microcontroller and than transmitted. At the receiver side, it is read into the flash of the receiving microcontroller. The microcontroller will be needed to set the address bit of the encoder and the transmission bit. The transmission bit will trigger the encoder to transmit the data.

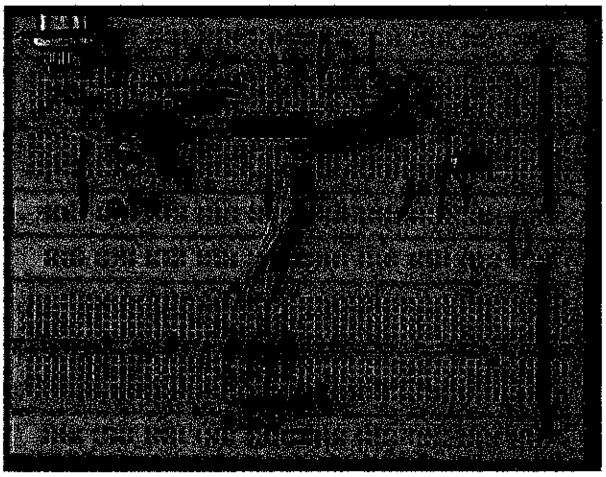


Figure 17 Transmitter circuit

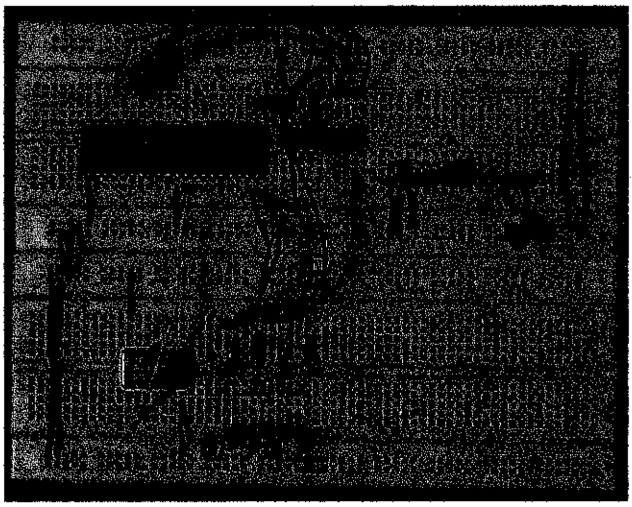


Figure 18 Receiver circuit

4.4.1 Transmitter Part

The role of the transmitter portion of the hardware is to take data from the serial port of a computer, modulate is as an amplitude shift keyed (ASK) signal in the 434 MHz band, and radiate the modulated signal via antenna. The circuit diagram is attached in APPENDIX J. The transmitter will work with serial communications commands built into the PIC compiler. Streams of bit that represent character value for PC will be sent to the communication pin of the microcontroller via RS232. MAX232 is than used to convert the data to readable form for transmission. Figure 19 below shows block diagram at the transmitter side.

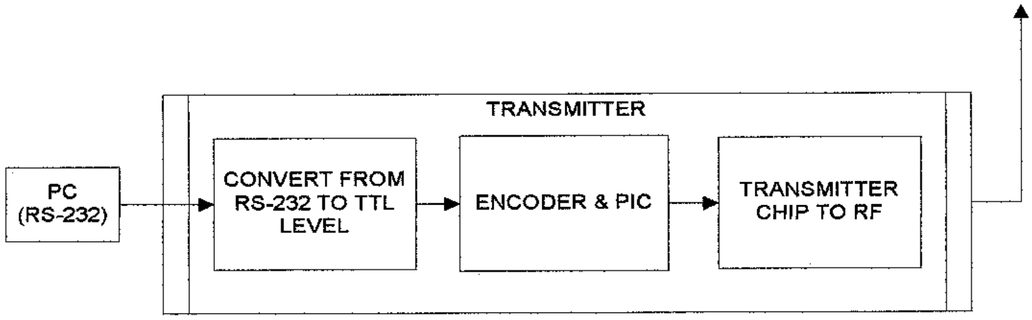


Figure 19 Transmitter block diagram

The program of communication microcontroller determines that state output that will be send via transmitter. C coding for transmitter part is attached in APPENDIX O.

4.4.2 Receiver Part

The receiver unit will detect and demodulate the ASK signal sent from the transmitter. Serial output from the receiver is than decoded so that desired parallel output can be obtained and sent to microcontroller. Pre programmed microcontroller will than determined the movement of the servo motor. Circuit diagram for receiver is attached in APPENDIX K and the block diagram is as shown in Figure 20. By using the circuit in APPENDIX K, the servomotors can be controlled directly by receiving data bit from the receiver which is than translated by microcontroller at the receiver side. C coding for receiver part is attached in APPENDIX P.

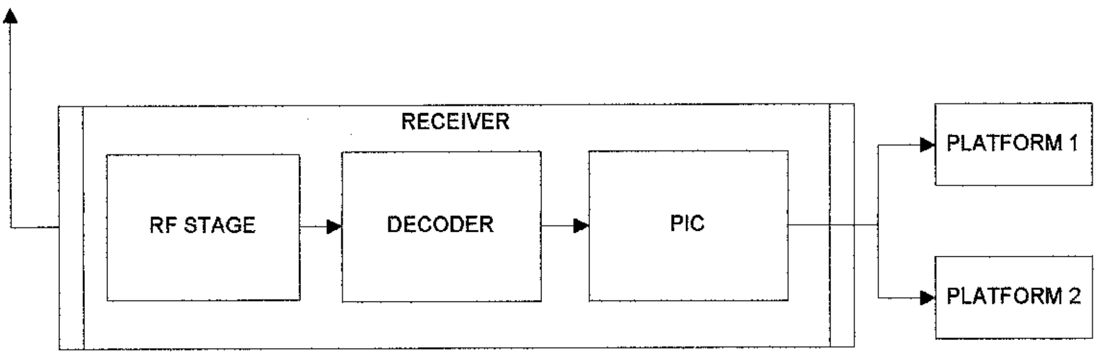


Figure 20 Receiver block diagram

4.5 Bluetooth Module

The second alternative solution to transmit data wirelessly is by using a Bluetooth Module. Bluetooth adepter is connected to the personal computer which will then provide a way to connect and send information to the bluetooth receiver module.

4.5.1 Bluetooth Basic Setup

Bluetooth module is power by 3.3V. A voltage regulator is needed so that to protect the module. High voltage will definitely damage the Bluetooth module while lower voltage will caused unstable operation. To regulate the voltage, LM1117 is used to supply 3.3V power to the module. Once the Bluetooth module is connected, it will appear in the main window below. The device is select to connect to serial port service by Interface using HyperTerminal. At the hyper terminal any number typed is displayed on the 7 segment. To control the servo motor, the module is interfaced with microcontroller. C coding is attached in APPENDIX Q. Circuit diagram for this part is attached in APPENDIX L.

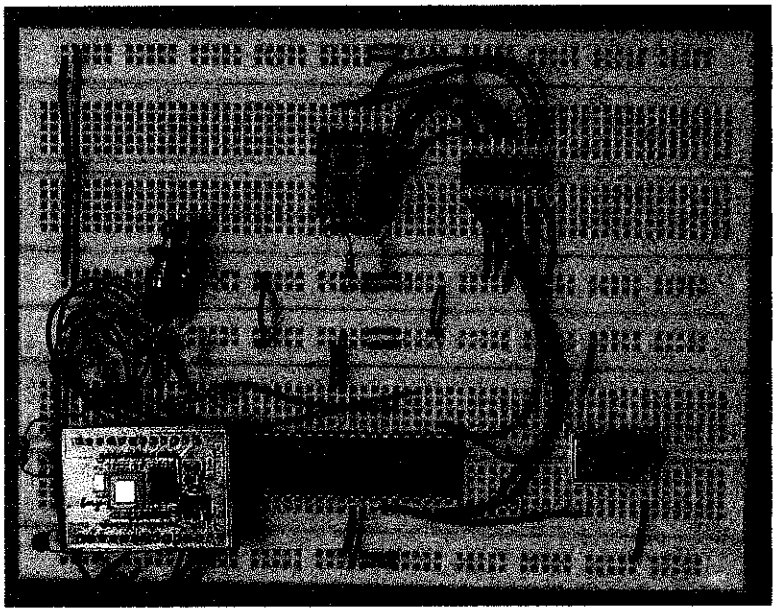


Figure 21 Circuit for Bluetooth data transmission

Once the Bluetooth dongle is inserted into an USB port and activated, by using BlueSoleil main window, it will search and discover any Bluetooth devices that are in discoverable mode. If the Bluetooth module is properly connected and powered up, it should be found by the software. Figure 22 below shows device found during the Bluetooth device search.

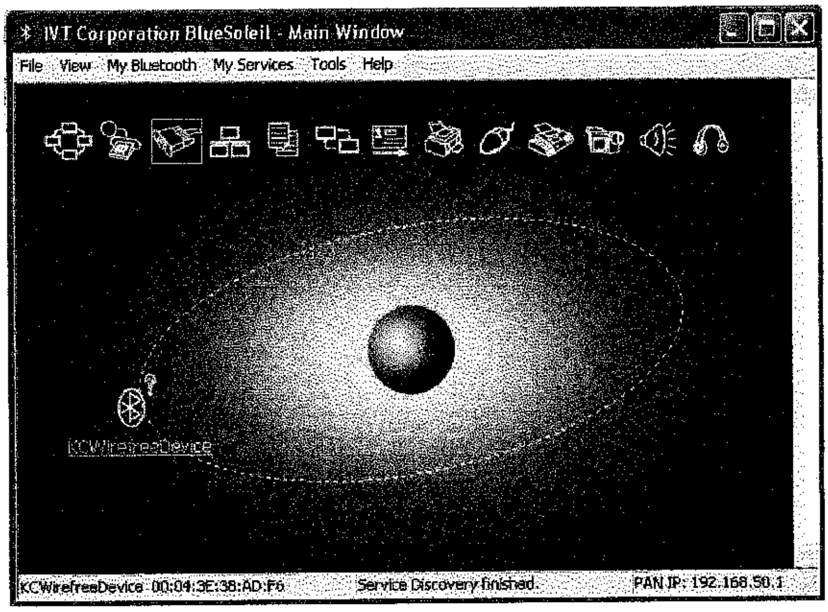


Figure 22 Method to connect with Bluetooth device

The symbol is selected to connect to the Bluetooth serial port service. Then, communication will be interface to the serial port service via hyper terminal. The Hyper terminal is connected to the serial port selection which depends on the serial port that was designated by BlueSoleil software. The appropriate baud rate is selected. Suitable baud rate for the Bluetooth module is 115200 bps.

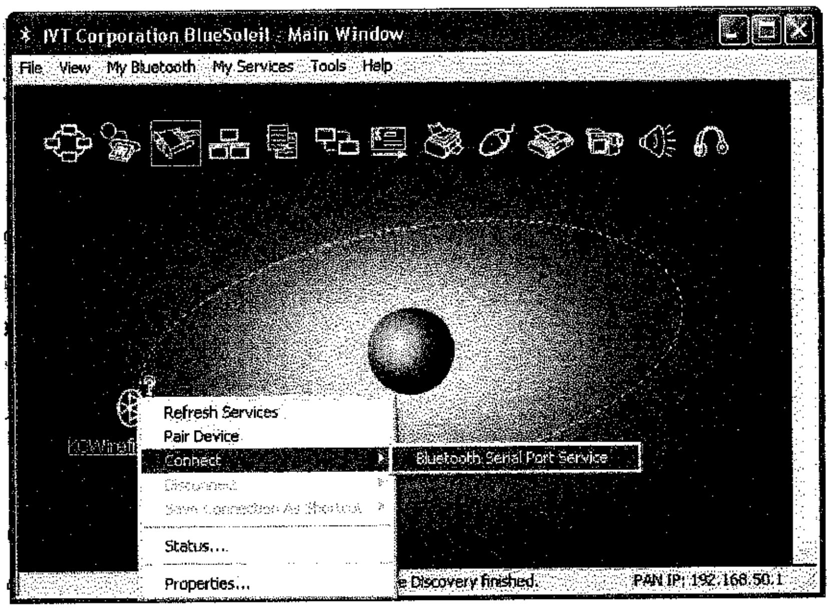


Figure 23 Bluetooth serial port setup

4.5.2 Bluetooth Connection to control Servo motor

Once the appropriate setup for Bluetooth connection is establish, HyperTerminal is ready to send data to control the movement of the servos. As for this project, four servos are controlled so that to have two unit of wireless pant and tilt camera platform. Two units of servos are used to cater for pant operation and two units for tilt operation. Figure 24 below shows how inputs from first microcontroller are sent to the second microcontroller.

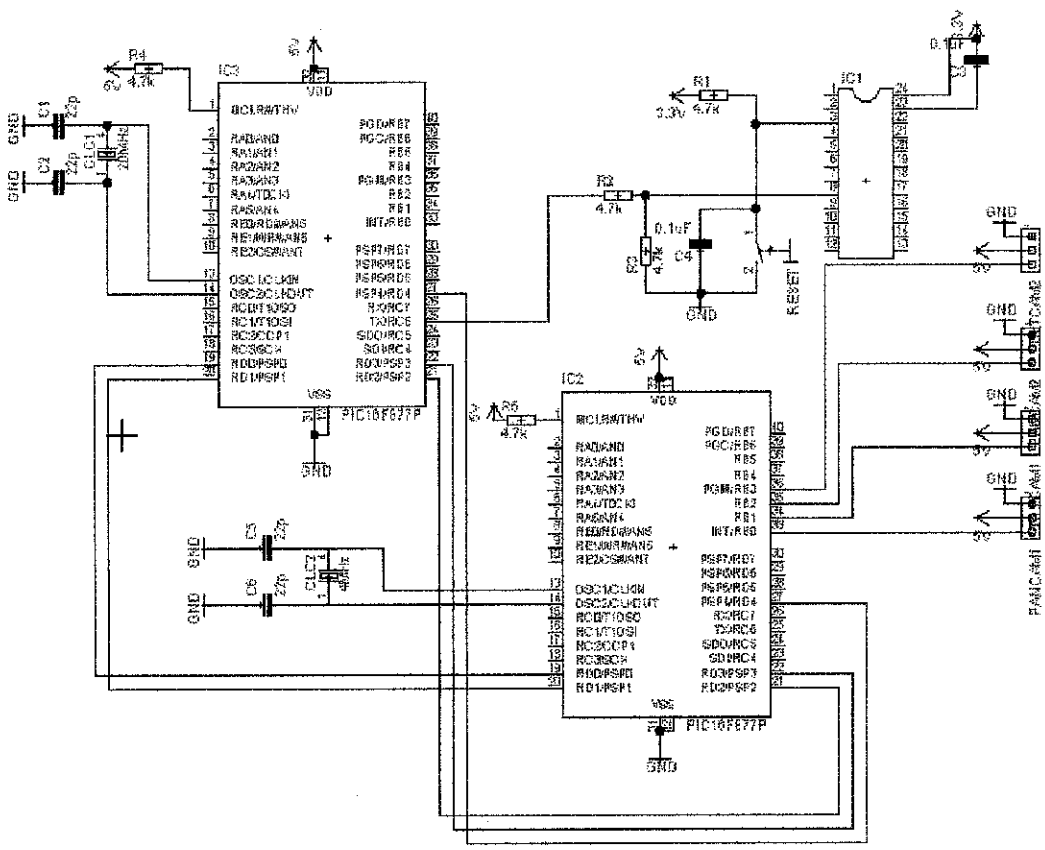


Figure 24 Schematic for Wireless Pan and Tilt Surveillance Platform

As we can see, the first microcontroller will receive input form the bluetooth module in the form of character and the output from the first microcontroller will become the input for the second microcontroller. C coding for the bluetooth wireless transmission is attached in APPENDIX R. The Character Translation for the first microcontroller is as shown in Table 4 below

Table 4 Character Translation Communication

Character	PORTD
1	00000001
2	00000010
3	00000011
4	00000100
5	00000101
6	00000110
7	00000111
8	00001000
9	00001001
0	00001010
q	00001011
w	00001100
e	00001101
r	00001110
t	00001111
y	00010000

Meanwhile, the second microcontroller is used to control the position for the cameras. Output pin B0, B1, B2 and B3 are used to generate PWM waveform according to input condition so that to control the movement for each servo. The PWM waveform is generated by creating a delay on the main loop of the PIC application. The choice of which output pin to use to drive the pulse hi and low is depends on the input from the keyboard. Table 5 below shows the respective character used to control the position of pan and tilt operation. C coding for the servo is attached in APPENDIX S.

Table 5 Servo controller Pin Operation

Character	Output Pin	Platform	Position
1	B0 high	PanCam1	90 degree (left)
2	B0 high	PanCam1	45 degree (left)
3	B0 high	PanCam1	0 degree (center)
4	B0 high	PanCam1	90 degree (right)
5	B1 high	PanCam1	45 degree (right)
6	B1 high	TiltCam1	45 degree (up)
7	B1 high	TiltCam1	0 degree (center)
8	B2 high	TiltCam1	45 degree (down)
9	B2 high	PanCam2	90 degree (left)
0	B2 high	PanCam2	45 degree (left)
q	B2 high	PanCam2	0 degree (center)
w	B2 high	PanCam2	90 degree (right)
e	B2 high	PanCam2	45 degree (right)
r	B3 high	TiltCam2	45 degree (up)
t	B3 high	TiltCam2	0 degree (center)
y	B3 high	TiltCam2	45 degree (down)

CHAPTER 5

CONCLUSION AND RECOMMENDATION

The main objective of this project is to have a wireless pan and tilt platform for object tracking utilizing webcam. The current wired system is using universal serial bus (USB) to send video signal from the webcam to the PC. From the data of object location, MATLAB will send the necessary commands to the platform to move along to the angle and direction of object location via serial port RS232. As for this project, RF and Bluetooth module is used to perform the wireless transmission part from PC so that to move the servo motors. The circuit design is test by sending character values from PC via hyper terminal into the communication pin of communication microcontroller. The task is focusing on sending of data from PC across the transmission module wirelessly. Bluetooth solution has a higher frequency and lower signal to noise ration compared to RF solution.

REFERENCES

[1] D.M. Bakker and Diane McMichael Gilster, Bluetooth End to End, Hungry Minds Inc, 2001

[2] http://en.wikipedia.org/wiki/Amplitude-shift_keying

[3] http://www.cs.ucl.ac.uk/staff/S.Bhatti/D51-notes/node12.html

[4] PIC 16F84A Datasheet, Microchip, Microchip Technology Inc, 2001

[5] HOLTEK 2^{12} Series of Encoders Datasheet, Microchip, Holter Semiconductor Inc, 1999

[6] HOLTEK 2^{12} Series of Decoder Datasheet, Microchip, Holter Semiconductor Inc, 1999

[7] PIC 16F877 Datasheet, Microchip, Microchip Technology Inc, 2001

[8] KC-21.3 Datasheet, Bluetooth OEM Module, KC Wirefree, 2006.05.21.

[9] TLP434A & RLP434A Datasheet, Laipac Technology, Inc.

[10] MAX232 Datasheet, Texas Instrument, 2002.

APPENDICES

APPENDIX A

BLUETOOTH KC-21 DATASHEET

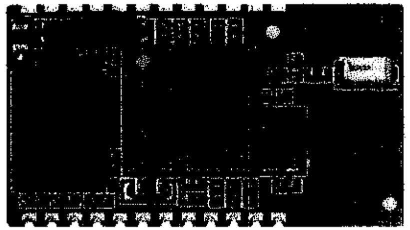
KC-21
Bluetooth OEM Module

Features

- Fully embedded Bluetooth v1.2 Serial Profile
- Class 2 radio
- Complete RF ready module
- Wireless data communications
- ARM7 microprocessor up to 48MHz
- 8Mb flash memory
- 921K baud data throughput
- Integrated chip antenna
- 128-bit encryption security
- Range up to 20m LOS
- SPI interface, up to 24MHz
- 14 general purpose I/O
- AT command set
- Multipoint capability
- FCC & Bluetooth qualified

Additional Documentation

- Getting Started Guide
- kcSerial 2.2 User Guide
- kcSerial 2.2 Reference Guide
- kcSerial 2.2 Configuration Guide



15mm x 27mm

Description

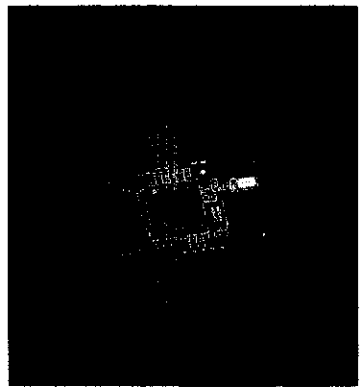
One of the most capable Bluetooth modules available, the KC-21 Bluetooth OEM Module is designed for maximum flexibility. The KC-21 module includes 14 general purpose input/output lines, and offers high speed serial communications up to 921K baud.

The KC-21 is a surface mount PCB module that provides fully embedded, ready to use Bluetooth wireless technology. The reprogrammable flash memory contains embedded firmware for serial cable replacement using the Bluetooth SPP profile. Other popular Bluetooth profiles are available.

We are able to quickly customize the firmware for external device interaction, or for optimizations such as minimal power consumption, high speed response, and other proprietary features. Custom firmware is easily pre-loaded into these highly tuned and tested modules so that they are ready to install without additional procedures.

Typical Cable Replacement Applications

- Serial communications
- Machine diagnostics and control
- Mobile financial transactions
- Remote sensing
- Medical device communications
- Industrial control
- Home automation



Software Architecture

Lower Layer Stack

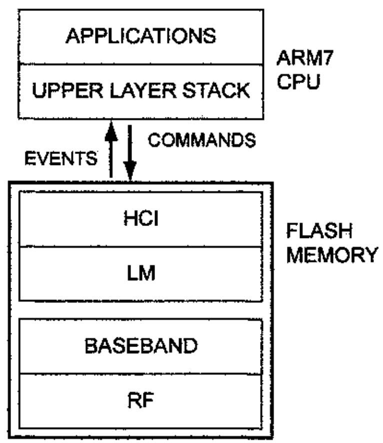
- Full Bluetooth data rate (723.3kbps maximum)
- Device power modes—active, sleep and deep sleep
- Wake on Bluetooth feature—optimized power consumption of host CPU
- Authentication and encryption
- Encryption key length from 8-bits to 128-bits maximum
- Persistent FLASH memory—for BD Address and radio parameter storage
- All ACL (Asynchronous Connection Less) packet types (DM1, DH1, DM3, DH3, DM5, DH5, AUX1)
- SCO (Synchronous Connection Oriented) packet types (HV1, HV2, HV3)
- Point to multipoint and scatternet support—3 master and 7 slave links allowed (10 active links simultaneously)
- Park, sniff, and hold modes—fully supported to maximum allowed intervals
- Master slave switch—supported during connection and post connection
- Dedicated Inquiry Access Code—for improved inquiry scan performance
- Dynamic packet selection—channel quality driven data rate to optimize link performance
- Dynamic power control—interference reduction and link performance
- Bluetooth test modes—per Bluetooth 1.2 specification
- 802.11b co-existence—AWMA and AFH
- SCO over UART, PCM, or SPI interface—application flexibility for host CPU
- Vendor specific HCI commands—to support device configuration and certification test modes

Upper Layer Stack

- SPP, SDAP, GAP, and DUN protocols
- RFComm, SDP, and L2CAP supported

HCI Interface

- Bluetooth 1.2 specification compliant
- HCI USB transport layer (H2)
- HCI UART transport layer (H4)
- Firmware upgrade over UART



HCI STACK

Hardware Specifications

General Conditions (V_{DD}= 3.3V and 25°C)

Recommended Operating Conditions

Rating	Min	Typical	Max	Unit
Operating Temperature Range	-25	-	80	°C
Supply Voltage V_{DD}	2.7	3.3	3.6	Volts
Signal Pin Voltage	-	3.3	-	Volts
RF Frequency	2400	-	2483.5	MHz

Absolute Maximum Ratings

Rating	Min	Typical	Max	Unit
Storage temperature range	-40	-	+150	°C
Supply voltage, V_{DD}	-0.3	-	+ 3.6	Volts
RF input power	-	-	-5	dBm

Current Consumption

Modes	Avg	Unit
Typical Power Consumption		
ACL data 115K Baud UART at max throughput (Master)	35.0	mA
ACL data 115K Baud UART at max throughput (Slave)	35.0	mA
Connection, no data traffic, master	18.0	mA
Connection, no data traffic, slave	28.0	mA
Connection in sniff (Tsniff=100ms), no data traffic, master	10.2	mA
Connection in sniff (Tsniff=100ms), no data traffic, slave	10.8	mA
Connection in sniff (Tsniff=375ms), no data traffic, master	2.75	mA
Connection in sniff (Tsniff=375ms), no data traffic, slave	3.50	mA
Standby, without deep sleep	16.5	mA
Standby, with deep sleep	0.16	mA
Page/Inquiry scan, deep sleep	2.1	mA
Peak current	90	mA

I/O Operating Characteristics

Symbol	Parameter	Min	Max	Unit	Conditions
V_{IL}	Low-Level Input Voltage	-	0.8	Volts	
V_{IH}	High-Level Input Voltage	2.0	-	Volts	
V_{OL}	Low-Level Output Voltage	-	0.4	Volts	I_{OL} = 2mA
V_{OH}	High-Level Output Voltage	2.4	-	Volts	I_{OH} = 2mA
I_{OL}	Low -Level Output Current	-	2.2	mA	V_{OL} = 0.4 V
I_{OH}	High-Level Output Current	-	3.1	mA	V_{OH} = 2.4 V
I_I	Input Leakage Current	-1	+1	µA	@V_I = 3.3V or 0V
V_{T+}	Schmitt Trigger Low-High	1.47	1.50	Volts	
V_{T-}	Schmitt Trigger High-Low	0.89	0.95	Volts	
R_{PU}	Pull-up Resistor	53	113	KΩ	Resistor Turned On
R_{PD}	Pull-down Resistor	43	118	KΩ	Resistor Turned On
C_I	Input Capacitance		7.5	pF	

USB I/O Operating Characteristics

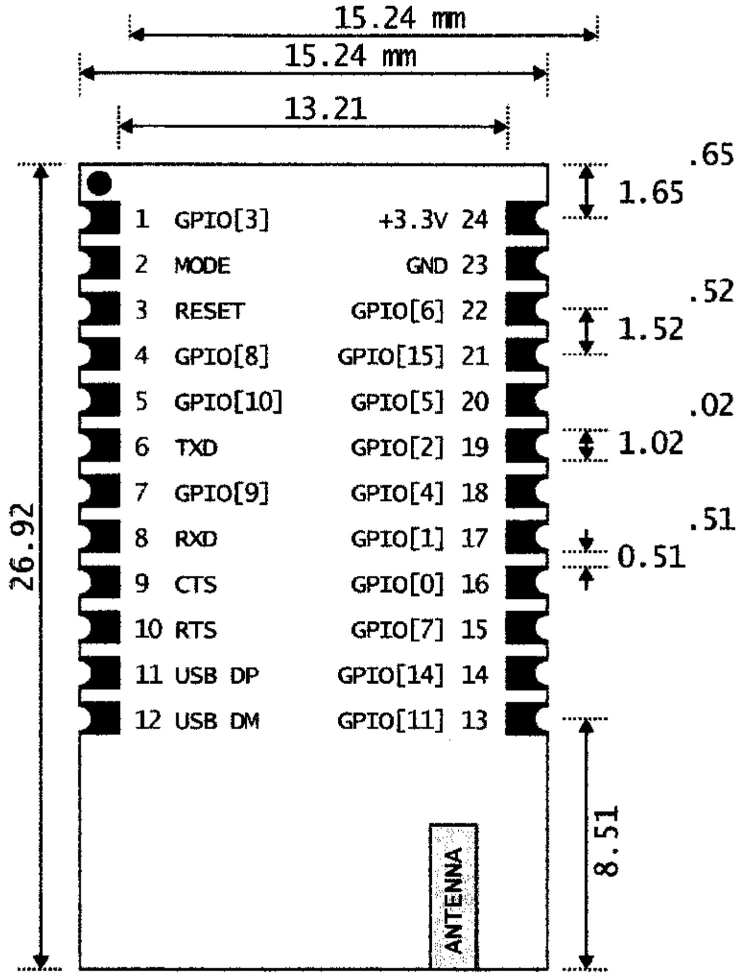
Symbol	Parameter	Min	Max	Unit	Conditions
V_{IL}	Low-Level Input Voltage	-	0.8	Volts	
V_{IH}	High-Level Input Voltage	2.0	-	Volts	
V_{OL}	Low-Level Output Voltage	-	0.3	Volts	
V_{OH}	High-Level Output Voltage	2.8	-	Volts	
C_{in}	Input Capacitance	-	25	pF	DP or DM to GND

Selected RF Characteristics

Parameters	Conditions	BT Spec	Typical	Unit
Antenna load			50	ohm
Radio Receiver				
Sensitivity level	BER < .001 with DH5	≤ -70	-85	dBm
Maximum usable level	BER < .001 with DH1	≥ -20	-5	dBm
Input VSWR			2.5:1	
Radio Transmitter				
Maximum output power	50 Ω load	-6 to +4	+1	dBm
Power control range		≥ 16	30	dB
Power control resolution		2 to 8	4	dB
Initial Carrier Frequency Tolerance		± 75	18	kHz
20 dB Bandwidth for modulated carrier		≤ 1000	930	kHz

Pin Assignment

Name	Type	Pin #	Description
UART Interface			
RXD	I	8	Receive data
TXD	O	6	Transmit data
CTS#	I	9	Clear to send (active low)
RTS#	O	10	Request to send (active low)
USB Interface			
DM	I/O	11	USB data minus
DP	I/O	12	USB data plus
Reserved			
MODE	I	2	Reserved
Power and Ground			
V_{DD}		24	V_{DD}
GND		23	GND
Reset			
RESET#	I	3	Reset input (active low for 5 ms); Schmidt triggered
GPIO – General Purpose Input/Output			
GPIO [0]	I/O	16	General Purpose Input/Output
GPIO [1]	I/O	17	General Purpose Input/Output
GPIO [2]	I/O	19	General Purpose Input/Output
GPIO [3]	I/O	1	General Purpose Input/Output
GPIO [4]	I/O	18	General Purpose Input/Output
GPIO [5]	I/O	20	General Purpose Input/Output
GPIO [6]	I/O	22	General Purpose Input/Output
GPIO [7]	I/O	15	General Purpose Input/Output
GPIO [8]	I/O	4	General Purpose Input/Output
GPIO [9]	I/O	7	General Purpose Input/Output
GPIO [10]	I/O	5	General Purpose Input/Output
GPIO [11]	I/O	13	General Purpose Input/Output
GPIO [14]	I/O	14	General Purpose Input/Output
GPIO [15]	I/O	21	General Purpose Input/Output



Height 3.10

Hardware Design

KC Wirefree modules support UART, USB, and GPIO hardware interfaces. This section details typical usage models for these features. Please note that the usage of these interfaces is dependant upon the firmware that is loaded into the module, and is beyond the scope of this document.

Notes

- RESET pin must be held high. 3.0v recommended.
- All unused pins should be left floating; do not ground.
- All GND pins must be well grounded.
- The area around the module should be free of any ground planes, power planes, trace routings, or metal for at least 8 mm from the antenna in all directions.
- Traces should not be routed underneath the module.
- The RTS pin is floating in this UART implementation. If the RTS pin is not connected, it must be pulled high using a 50K ohm resistor.

Module Reflow Installation

The KC-21 is a surface mount Bluetooth module supplied on a 24 pin, 6-layer PCB. For non Pb-free applications, Sn63Pb37 solder is recommended. The final assembly recommended reflow profile is:

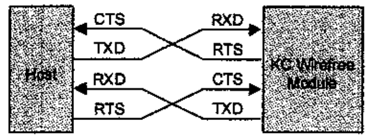
- Maximum peak temperature of 208° - 210°C (below 220°C).
- Maximum rise and fall slope after liquidous of < 2°C/second.
- Maximum rise and fall slope after liquidous of < 2°C/second.
- Maximum time at liquidous of 50 – 90 seconds.

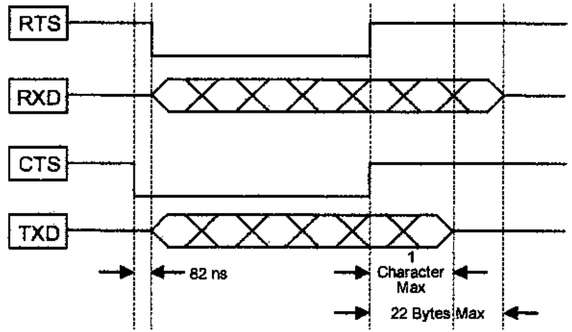
GPIO Interface

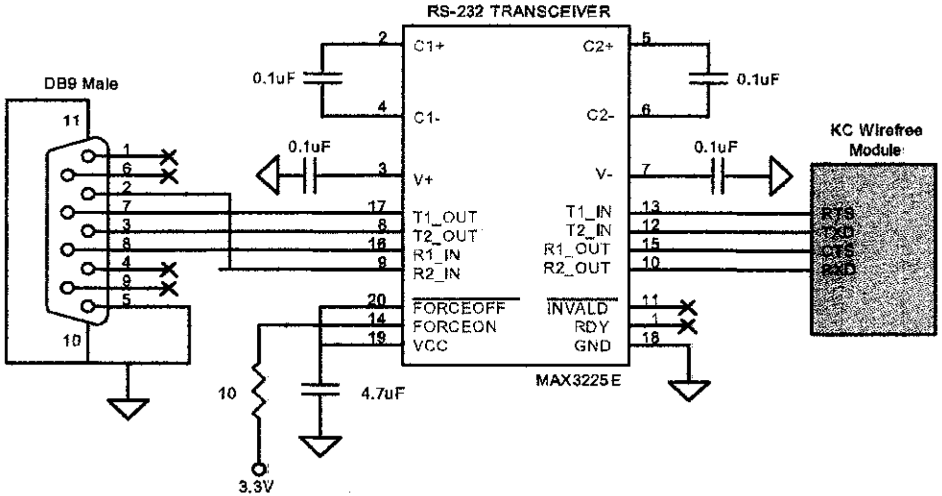
All GPIOs are capable of sinking and sourcing 2mA of I/O current. These terminals are 5V tolerant. GPIO [0] to GPIO [7] are internally pulled down with 50KΩ (nominal) resistors GPIO [8] to GPIO [15] are internally pulled up with 50KΩ (nominal) resistors.

UART Interface

The UART is compatible with the 16450 industry standard. Four signals are provided with the UART interface. The TXD and RXD pins are used for data while the CTS and RTS pins are used for flow control.







FCC Regulatory Compliance

This module has been tested and found to comply with the FCC Part15 Rules. These limits are designed to provide reasonable protection against harmful interference in approved installations. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Modifications or changes to this equipment not expressly approved by KC Wirefree may void the user's authority to operate this equipment.

Limited Modular Approval

FCC ID: S22BTMODULE-CL2

In accordance with FCC Part 15, the KC-21 is listed above as a Limited Modular Transmitter device.

In support of the Modular Transmitter Approval, the following is stated:

- The module does have buffered modulation / data inputs.
- The module does regulate its own power supply.
- The module does have a permanently attached antenna.
- The module can be tested as a stand-alone device.
- The module is labeled with the proper FCC ID, and labeling instructions are provided to OEM end users for external product labels.
- The module does have instruction for proper use.
- The module does meet the FCC RF regulations.

Limited Modular Transmitter Approval, is granted, instead of Modular Transmitter Approval, because the following condition is not met:

- The module does not have its own RF shielding.

In accordance with FCC document, DA 00-1407, all final usage by OEMs of this device, (1) Must be approved by the module's manufacturer, KC Wirefree, (2) final installation must follow the instructions in this user's manual, (3) a written agreement with the OEM will detail which products are approved for this module's final installation to control its end usage and ensure FCC Part 15 compliance.

FCC Label Instructions

The outside of final products that contain a KC-21 device must display a label referring to the enclosed module. This exterior label can use wording such as the following: "Contains Transmitter Module FCC ID: S22BTMODULE-CL2" or "Contains FCC ID: S22BTMODULE-CL2". Any similar wording that expresses the same meaning may be used.

Firmware Reprogramming Notes

KC Wirefree modules and adaptors may be reprogrammed using the procedure detailed in this document. Normally, KC Wirefree will deliver devices with the final firmware already pre-loaded, so this should not need to be part of a product's normal production.

Typically, the following tools and equipment are necessary:

- New flash image and KC Wirefree Flash Tool.
- Null modem serial cable to PC serial port (available from KC Wirefree).
- TTL to RS232 level shifter; when programming modules.

UART Connections

The new firmware must be loaded using the UART interface. Access to this interface depends on the product type.

The minimum UART pins that are needed are the Rx and Tx pins. The RTS and CTS flow control pins are not necessary for programming devices at 115K baud (standard), but are required at 921K baud.

Serial Adaptors

KC-111 and KC-121 serial adaptors already have a DB9 serial connector and RS232 level output, so no additional hardware modifications or connections are necessary.

Modules

KC-11, KC-20, KC-21, and KC-22 modules support TTL level UART communications, but a PC requires RS232 voltage levels. Therefore, a TTL to RS232 level shifter is required.

Flash Tool

KC Wirefree will deliver the appropriate firmware in a directory structure that includes the flash tool batch (.bat) files and firmware image. The following procedure must be carefully followed to program the device:

1. Make the necessary serial connections to the PC.
2. Start with the device powered OFF.
3. Run the BurnFirmware_pX.bat file, where X represents the COM port, 1-4, of the PC connection.
4. A DOS shell will appear, and prompt the user to reset the device.
5. Turn on power to the device.
6. The program will automatically establish communication and loads the new firmware in about one minute.

Multipoint Cable Replacement

KC Wirefree modules may be loaded with point-to-multipoint firmware which supports up to 4 simultaneous connections. Only one master link is supported.

Protocol Summary

Since multiple devices may be communicating at the same time, a simple serial protocol was added in order to direct the data to the desired node; a layer 2 protocol on top of the passthrough mode layer.

This protocol consists of a destination/source field and a length field:

Dest/Source Node:
One ASCII digit (1 byte) indicating Node ID [0-9]

Length:
Three ASCII digits (3 bytes) indicating data length [001-315]

Data:
Up to 315 bytes

AT Commands

Two AT commands, connect and disconnect, were modified for this sample application as follows:

```
AT+ZV SppConnect [remote address]
```

This command has the same syntax as before, but a Node Id for the remote device is automatically assigned when the connection is successfully established. These node assignments begin at 0, and will increment as new connections are added using successive SppConnect commands. The lowest available Node Id will always be assigned next, in cases where disconnections have caused an Id to become available. Node Ids cannot be manually assigned in this application.

When the device is connected to, as a slave (client), a Node Id is also assigned to the initiating device independently. Thus, the devices forming a new connection may not assign each other the same Node Id.

```
AT+ZV SppDisconnect [remote Node Id]
```

A Node Id parameter has been added to the disconnect command. Only 1 device may be disconnected per command.

Bypass and Command modes are basically the same as in the standard point-to-point application.

Example

Device A initiates a connection to device B:

```
AT+ZV SppConnect [address of B]
```

A Node Id of 0 is assigned to B for A's network.

B assigns A an Id of 0 as well.

A sends data to B as follows (bypass mode): `0004abcd`

Note: no Cr or Lf is necessary

B displays: `0004abcd`

A initiates a connection to device C: (command mode)

```
AT+ZV SppConnect [address of C]
```

A Node Id of 1 is assigned to C in A's network.

C assigns A an Id of 0.

A sends data to C as follows (bypass mode): `1003efg`

C displays: `0003efg`

C sends data to A as follows: `0003xyz`

A displays: `1003xyz`

Note: the Node Ids do not match.

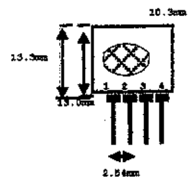
B and C cannot send data to each other.

APPENDIX B

TLP-434A/RLP-434A DATASHEET

TLP434A Ultra Small Transmitter

Easy-Link Wireless



pin 1 : GND
pin 2 : Data In
pin 3 : Vcc
pin 4 : Antenna (RF output)

Frequency 315, 418 and 433.92 Mhz

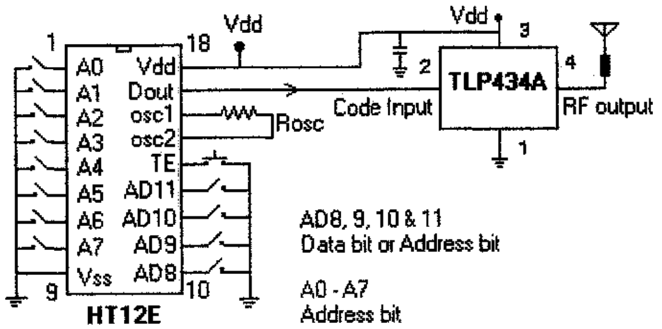
Modulation : ASK
Operation Voltage : 2 - 12 VDC

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Vcc	Operating supply voltage		2.0	-	12.0	V
Icc 1	Peak Current (2V)		-	-	1.64	mA
Icc 2	Peak Current (12V)		-	-	19.4	mA
Vh	Input High Voltage	Idata= 100uA (High)	Vcc-0.5	Vcc	Vcc+0.5	V
Vl	Input Low Voltage	Idata= 0 uA (Low)	-	-	0.3	V
FO	Absolute Frequency	315Mhz module	314.8	315	315.2	MHz
PO	RF Output Power- 50ohm	Vcc = 9V-12V	-	16	-	dBm
		Vcc = 5V-6V	-	14	-	dBm
DR	Data Rate	External Encoding	512	4.8K	200K	bps

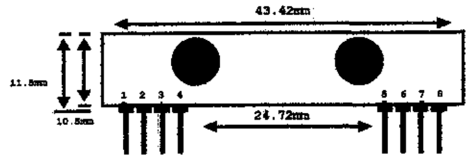
Notes : (Case Temperature = 25°C +- 2°C , Test Load Impedance = 50 ohm)

Application Circuit :

Typical Key-chain Transmitter using HT12E-18DIP, a Binary 12 bit Encoder from Holtek Semiconductor Inc.



RLP434A SAW Based Receiver



pin 1 : Gnd
pin 2 : Digital Data Output
pin 3 : Linear Output /Test
pin 4 : Vcc
pin 5 : Vcc
pin 6 : Gnd
pin 7 : Gnd
pin 8 : Antenna

Frequency 315, 418 and 433.92 Mhz

Modulation : ASK
Supply Voltage : 3.3 - 6.0 VDC
Output : Digital & Linear

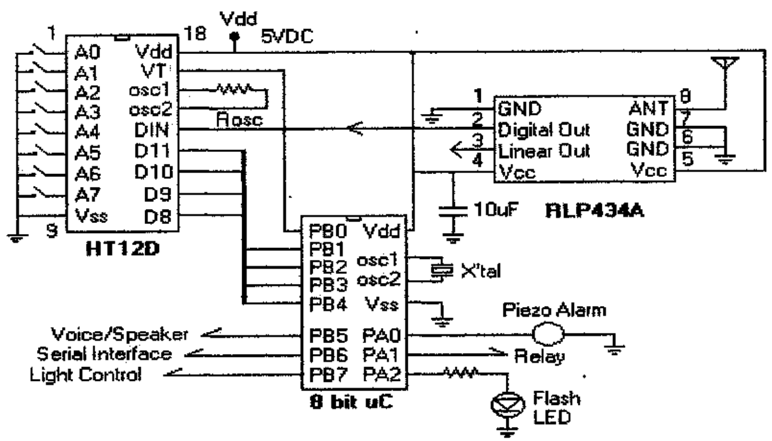
Symbol	Parameter	Conditions	Min	Typ	Max	
Vcc	Operating supply voltage		3.3	5.0V	6.0	V
Itot	Operating Current		-	4.5		mA
Vdata	Data Out	Idata = +200 uA (High)	Vcc-0.5	-	Vcc	V
		Idata = -10 uA (Low)	-	-	0.3	V

Electrical Characteristics

Characteristics	SYM	Min	Typ	Max	Unit
Operation Radio Frequency	FC	315, 418 and 433.92			MHz
Sensitivity	Pref		-110		dBm
Channel Width			+-500		Khz
Noise Equivalent BW			4		Khz
Receiver Turn On Time			5		ms
Operation Temperature	Top	-20	-	80	C
Baseboard Data Rate			4.8		KHz

Application Circuit :

Typical RF Receiver using HT12D-18DIP, a Binary 12 bit Decoder with 8 bit uC HT48RXX from Holtek Semiconductor Inc.



Laipac Technology, Inc.
55 West Beaver Creek Rd. Unit 1 Richmond Hill Ontario L4B 1K5 Canada
Tel: (905)762-1228 Fax: (905)763-1737 e-mail: info@laipac.com

LAIPAC TECH

APPENDIX C

HT-12E ENCODER DATASHEET

HOLTEK

2^{12} Series of Encoders

Features

- Operating voltage
  - 2.4V~5V for the HT12A
  - 2.4V~12V for the HT12E
- Low power and high noise immunity CMOS technology
- Low standby current: 0.1μA (typ.) at V_{DD}=5V
- HT12A with a 38kHz carrier for infrared transmission medium
- Minimum transmission word
  - Four words for the HT12E
  - One word for the HT12A
- Built-in oscillator needs only 5% resistor
- Data code has positive polarity
- Minimal external components
- 18-pin DIP or 20-pin SOP package available for HT12A
- 14/18-pin DIP or 16/20-pin SOP or 16-pin NSOP package available for HT12E

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 2^{12} encoders are a series of CMOS LSIs for remote control system applications. They are capable of encoding information which consists of N address bits and 12–N data bits. Each address/data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header bits via an RF or an infrared transmission medium upon receipt of a trigger signal. The capability to select a $\overline{TE}$ trigger on the HT12E or a DATA trigger on the HT12A further enhances the application flexibility of the 2^{12} series of encoders. The HT12A additionally provides a 38kHz carrier for infrared systems.

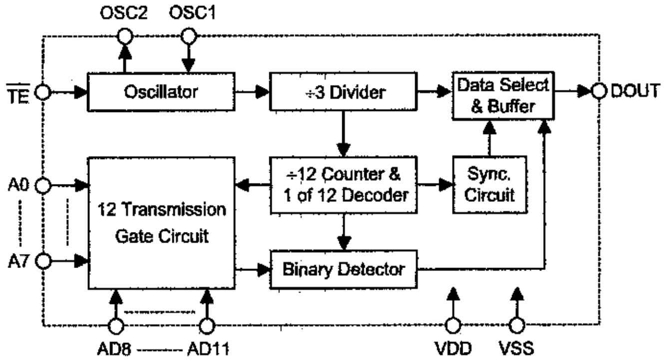
Selection Table

Function / Part No.	Address No.	Address/ Data No.	Data No.	Oscillator	Trigger	Package	Carrier Output	Negative Polarity
HT12A	8	0	4	455kHz resonator	D8~D11	18 DIP 20 SOP	38kHz	No
HT12E	8	4	0	RC oscillator	$\overline{TE}$	14/18 DIP 16/20 SOP 16 NSOP	No	No

Block Diagram

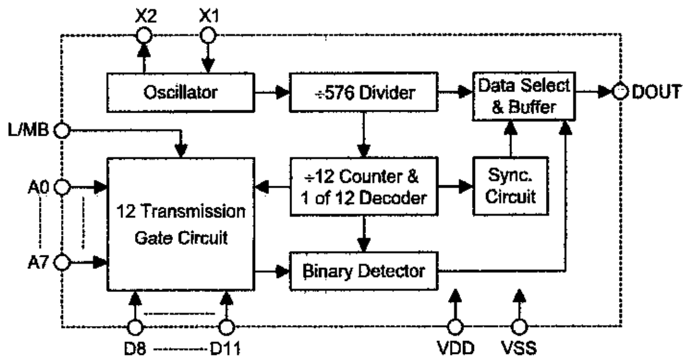
$\overline{\text{TE}}$ trigger

HT12E



DATA trigger

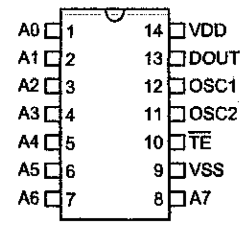
HT12A



Note: The address data pins are available in various combinations (refer to the address/data table).

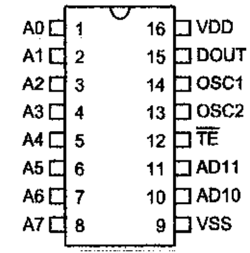
Pin Assignment

8-Address



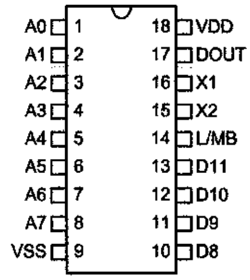
HT12E
-14 DIP

8-Address
2-Address/Data



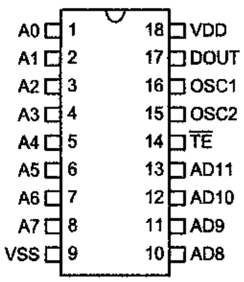
HT12E
-16 SOP

8-Address
4-Data



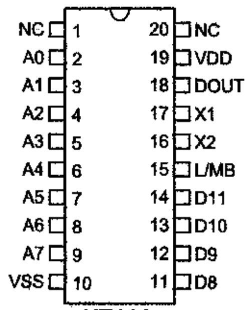
HT12A
-18 DIP

8-Address
4-Address/Data



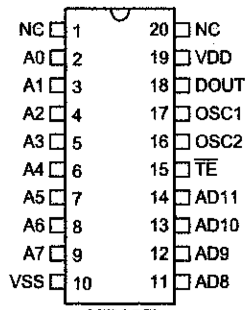
HT12E
-18 DIP

8-Address
4-Data



HT12A
- 20 SOP

8-Address
4-Address/Data



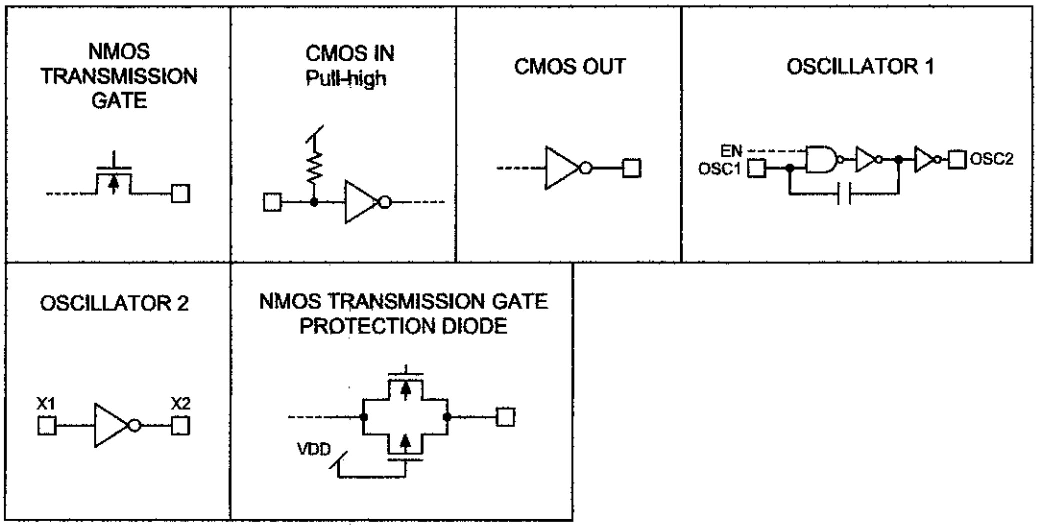
HT12E
- 20 SOP

Pin Description

Pin Name	I/O	Internal Connection	Description
A0~A7	I	CMOS IN Pull-high (HT12A) NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address A0~A7 setting These pins can be externally set to VSS or left open
AD8~AD11	I	NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address/data AD8~AD11 setting These pins can be externally set to VSS or left open
D8~D11	I	CMOS IN Pull-high	Input pins for data D8~D11 setting and transmission enable, active low These pins can be externally set to VSS or left open (see Note)
DOUT	O	CMOS OUT	Encoder data serial transmission output
L/MB	I	CMOS IN Pull-high	Latch/Momentary transmission format selection pin: Latch: Floating or VDD Momentary: VSS
$\overline{\text{TE}}$	I	CMOS IN Pull-high	Transmission enable, active low (see Note)
OSC1	I	OSCILLATOR 1	Oscillator input pin
OSC2	O	OSCILLATOR 1	Oscillator output pin
X1	I	OSCILLATOR 2	455kHz resonator oscillator input
X2	O	OSCILLATOR 2	455kHz resonator oscillator output
VSS	I	—	Negative power supply (GND)
VDD	I	—	Positive power supply

Notes: D8~D11 are all data input and transmission enable pins of the HT12A.
$\overline{\text{TE}}$ is a transmission enable pin of the HT12E.

Approximate internal connections



Absolute Maximum Ratings

Supply Voltage (HT12A) ....–0.3V to 5.5V

Input Voltage....................V_{SS}–0.3 to V_{DD}+0.3V

Operating Temperature...............–20°C to 75°C

Supply Voltage (HT12E) ....–0.3V to 13V

Storage Temperature.................–50°C to 125°C

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

Electrical Characteristics

HT12A

Ta=25°C

Symbol	Parameter	Test Conditions V_{DD}	Test Conditions Conditions	Min.	Typ.	Max.	Unit
V_{DD}	Operating Voltage	—	—	2.4	3	5	V
I_{STB}	Standby Current	3V	Oscillator stops	—	0.1	1	μA
		5V		—	0.1	1	μA
I_{DD}	Operating Current	3V	No load f_{OSC}=455kHz	—	200	400	μA
		5V		—	400	800	μA
I_{DOUT}	Output Drive Current	5V	$V_{OH}=0.9V_{DD}$ (Source)	–1	–1.6	—	mA
			$V_{OL}=0.1V_{DD}$ (Sink)	2	3.2	—	mA
V_{IH}	"H" Input Voltage	—	—	$0.8V_{DD}$	—	V_{DD}	V
V_{IL}	"L" Input Voltage	—	—	0	—	$0.2V_{DD}$	V
R_{DATA}	D2~D11 Pull-high Resistance	5V	V_{DATA}=0V	—	150	300	kΩ

HT12E

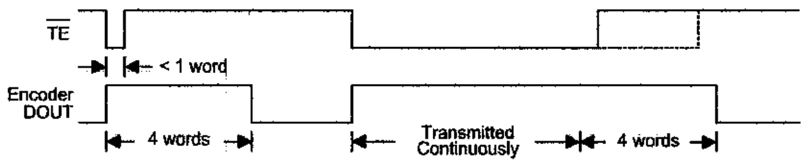
Ta=25°C

Symbol	Parameter	Test Conditions V_{DD}	Test Conditions Conditions	Min.	Typ.	Max.	Unit
V_{DD}	Operating Voltage	—	—	2.4	5	12	V
I_{STB}	Standby Current	3V	Oscillator stops	—	0.1	1	μA
		12V		—	2	4	μA
I_{DD}	Operating Current	3V	No load f_{OSC}=3kHz	—	40	80	μA
		12V		—	150	300	μA
I_{DOUT}	Output Drive Current	5V	$V_{OH}=0.9V_{DD}$ (Source)	–1	–1.6	—	mA
			$V_{OL}=0.1V_{DD}$ (Sink)	1	1.6	—	mA
V_{IH}	"H" Input Voltage	—	—	$0.8V_{DD}$	—	V_{DD}	V
V_{IL}	"L" Input Voltage	—	—	0	—	$0.2V_{DD}$	V
f_{OSC}	Oscillator Frequency	5V	R_{OSC}=1.1MΩ	—	3	—	kHz
$R_{\overline{TE}}$	$\overline{TE}$ Pull-high Resistance	5V	$V_{\overline{TE}}$=0V	—	1.5	3	MΩ

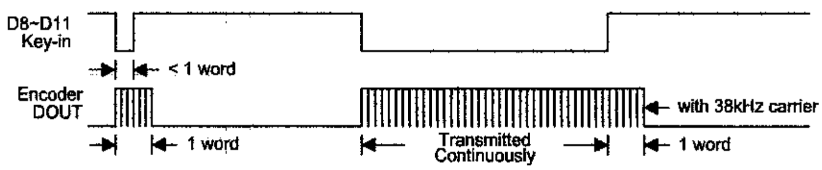
Functional Description

Operation

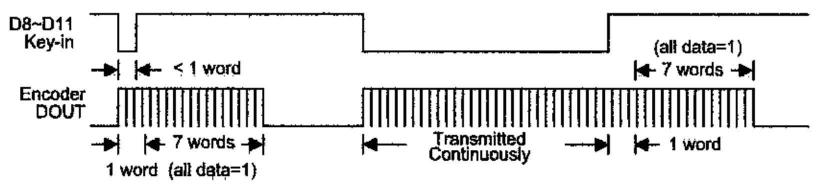
The 2^{12} series of encoders begin a 4-word transmission cycle upon receipt of a transmission enable ($\overline{TE}$ for the HT12E or D8~D11 for the HT12A, active low). This cycle will repeat itself as long as the transmission enable ($\overline{TE}$ or D8~D11) is held low. Once the transmission enable returns high the encoder output completes its final cycle and then stops as shown below.



Transmission timing for the HT12E



Transmission timing for the HT12A (L/MB=Floating or VDD)

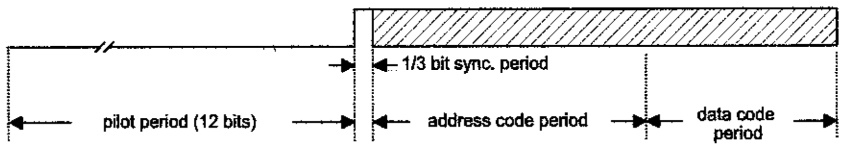


Transmission timing for the HT12A (L/MB=VSS)

Information word

If L/MB=1 the device is in the latch mode (for use with the latch type of data decoders). When the transmission enable is removed during a transmission, the DOUT pin outputs a complete word and then stops. On the other hand, if L/MB=0 the device is in the momentary mode (for use with the momentary type of data decoders). When the transmission enable is removed during a transmission, the DOUT outputs a complete word and then adds 7 words all with the "1" data code.

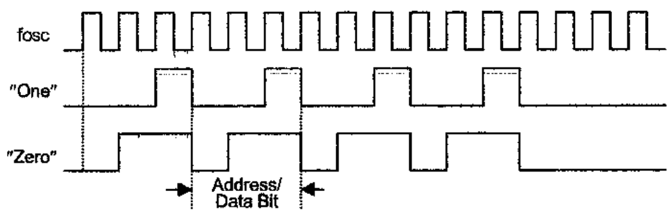
An information word consists of 4 periods as illustrated below.



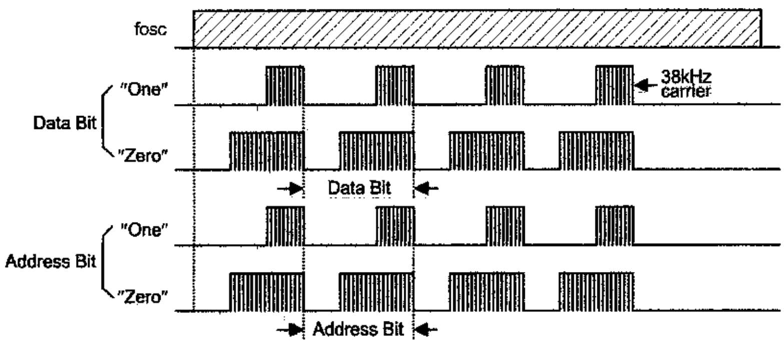
Composition of information

Address/data waveform

Each programmable address/data pin can be externally set to one of the following two logic states as shown below.



Address/Data bit waveform for the HT12E



Address/Data bit waveform for the HT12A

The address/data bits of the HT12A are transmitted with a 38kHz carrier for infrared remote controller flexibility.

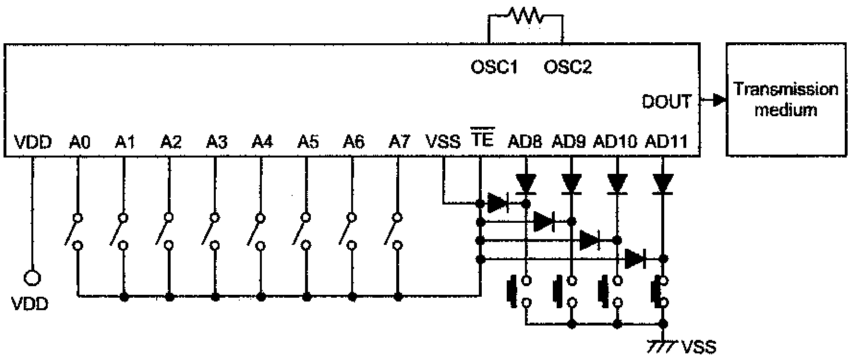
Address/data programming (preset)

The status of each address/data pin can be individually pre-set to logic "high" or "low". If a transmission-enable signal is applied, the encoder scans and transmits the status of the 12 bits of address/data serially in the order A0 to AD11 for the HT12E encoder and A0 to D11 for the HT12A encoder.

During information transmission these bits are transmitted with a preceding synchronization bit. If the trigger signal is not applied, the chip enters the standby mode and consumes a reduced current of less than 1μA for a supply voltage of 5V.

Usual applications preset the address pins with individual security codes using DIP switches or PCB wiring, while the data is selected by push buttons or electronic switches.

The following figure shows an application using the HT12E:



The transmitted information is as shown:

Pilot & Sync.	A0	A1	A2	A3	A4	A5	A6	A7	AD8	AD9	AD10	AD11
	1	0	1	0	0	0	1	1	1	1	1	0

Address/Data sequence

The following provides the address/data sequence table for various models of the 2^{12} series of encoders. The correct device should be selected according to the individual address and data requirements.

Part No.	Address/Data Bits											
	0	1	2	3	4	5	6	7	8	9	10	11
HT12A	A0	A1	A2	A3	A4	A5	A6	A7	D8	D9	D10	D11
HT12E	A0	A1	A2	A3	A4	A5	A6	A7	AD8	AD9	AD10	AD11

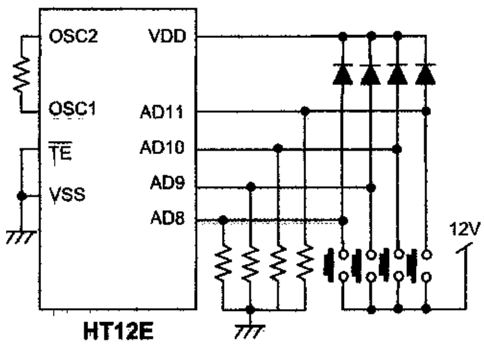
Transmission enable

For the HT12E encoders, transmission is enabled by applying a low signal to the $\overline{TE}$ pin. For the HT12A encoders, transmission is enabled by applying a low signal to one of the data pins D8~D11.

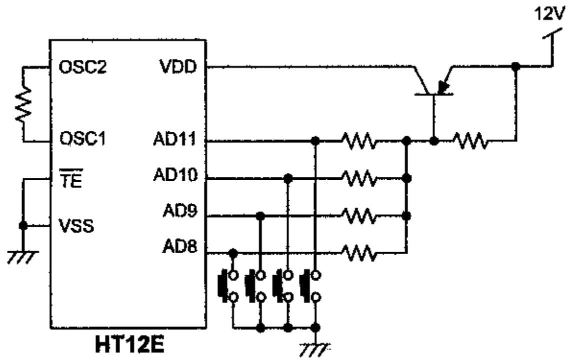
Two erroneous HT12E application circuits

The HT12E must follow closely the application circuits provided by Holtek (see the "Application circuits").

- Error: AD8~AD11 pins input voltage > V_{DD}+0.3V

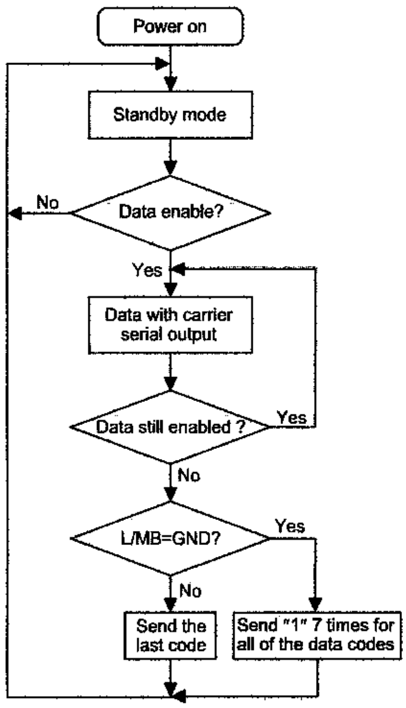


• Error: The IC's power source is activated by pins AD8~AD11

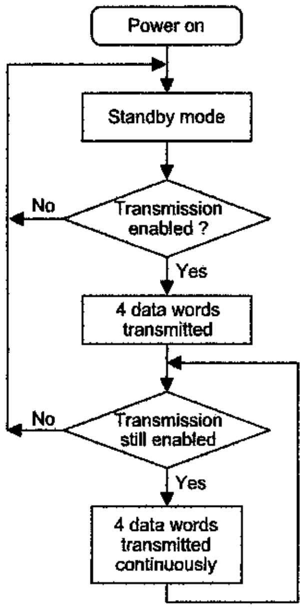


Flowchart

• HT12A

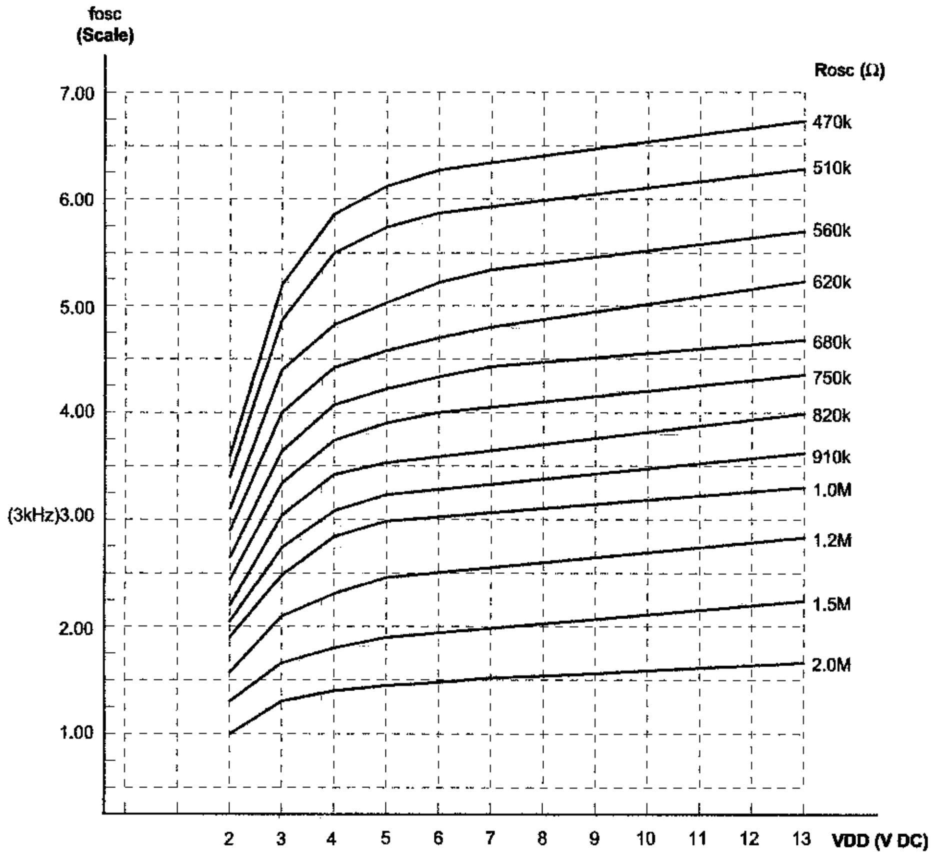


• HT12E



Notes: D8~D11 are transmission enables of the HT12A.
$\overline{TE}$ is the transmission enable of the HT12E.

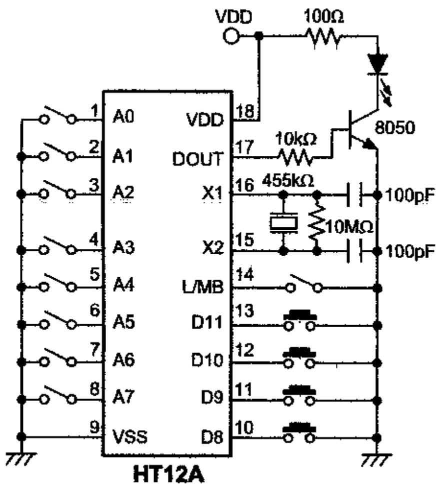
Oscillator frequency vs supply voltage

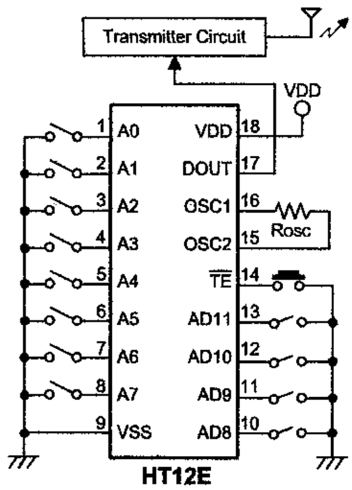


The recommended oscillator frequency is f_{OSCD} (decoder) $\cong 50\ f_{OSCE}$ (HT12E encoder)

$\cong \frac{1}{3} f_{OSCE}$ (HT12A encoder)

Application Circuits





Notes: Typical infrared diode: EL-1L2 (KODENSHI CORP.)
Typical RF transmitter: JR-220 (JUWA CORP.)

APPENDIX D

HT-12D DECODER DATASHEET

HOLTEK

2^{12} Series of Decoders

Features

- Operating voltage: 2.4V~12V
- Low power and high noise immunity CMOS technology
- Low standby current
- Capable of decoding 12 bits of information
- Pair with Holtek's 2^{12} series of encoders
- Binary address setting
- Received codes are checked 3 times
- Address/Data number combination
  - HT12D: 8 address bits and 4 data bits
  - HT12F: 12 address bits only
- Built-in oscillator needs only 5% resistor
- Valid transmission indicator
- Easy interface with an RF or an infrared transmission medium
- Minimal external components

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 2^{12} decoders are a series of CMOS LSIs for remote control system applications. They are paired with Holtek's 2^{12} series of encoders (refer to the encoder/decoder cross reference table). For proper operation, a pair of encoder/decoder with the same number of addresses and data format should be chosen.

The decoders receive serial addresses and data from a programmed 2^{12} series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. They compare the serial input data three times continuously with their local addresses. If no error or unmatched codes are found, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission.

The 2^{12} series of decoders are capable of decoding informations that consist of N bits of address and 12–N bits of data. Of this series, the HT12D is arranged to provide 8 address bits and 4 data bits, and HT12F is used to decode 12 bits of address information.

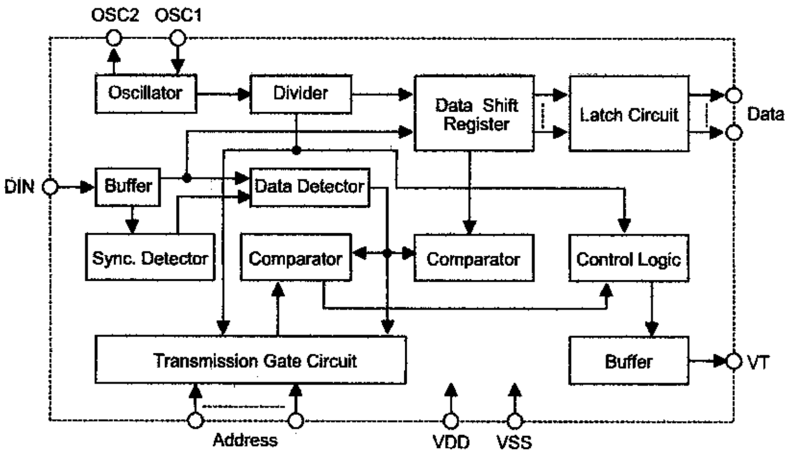
Selection Table

Function / Part No.	Address No.	Data No.	Data Type	VT	Oscillator	Trigger	Package
HT12D	8	4	L	√	RC oscillator	DIN active "Hi"	18 DIP/20 SOP
HT12F	12	0	—	√	RC oscillator	DIN active "Hi"	18 DIP/20 SOP

Notes: Data type: L stands for latch type data output.

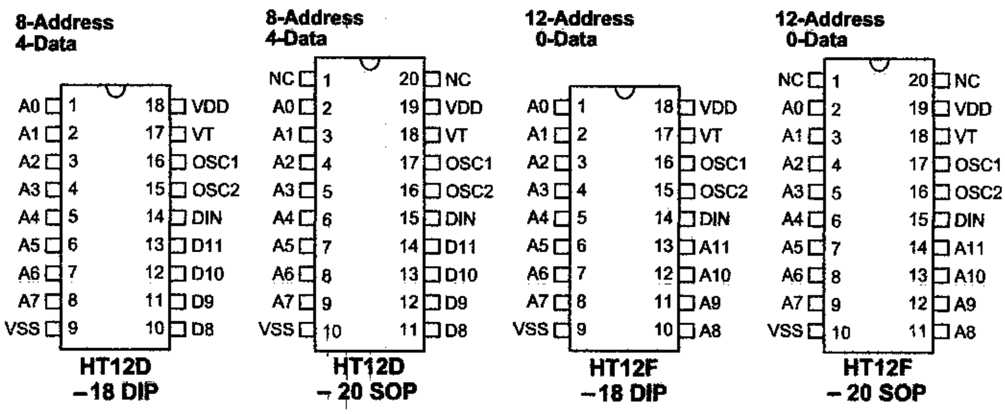
VT can be used as a momentary data output.

Block Diagram



Note: The address/data pins are available in various combinations (see the address/data table).

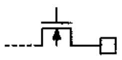
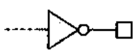
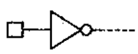
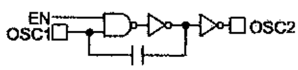
Pin Assignment



Pin Description

Pin Name	I/O	Internal Connection	Description
A0~A11	I	NMOS TRANSMISSION GATE	Input pins for address A0~A11 setting They can be externally set to VDD or VSS.
D8~D11	O	CMOS OUT	Output data pins
DIN	I	CMOS IN	Serial data input pin
VT	O	CMOS OUT	Valid transmission, active high
OSC1	I	OSCILLATOR	Oscillator input pin
OSC2	O	OSCILLATOR	Oscillator output pin
VSS	I	—	Negative power supply (GND)
VDD	I	—	Positive power supply

Approximate internal connection circuits

NMOS TRANSMISSION GATE	CMOS OUT	CMOS IN	OSCILLATOR
			

Absolute Maximum Ratings

Supply Voltage................................–0.3V to 13V

Input Voltage....................V_{SS}–0.3 to V_{DD}+0.3V

Storage Temperature.................–50°C to 125°C

Operating Temperature ....–20°C to 75°C

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

Electrical Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	5	12	V
I_{STB}	Standby Current	5V	Oscillator stops	—	0.1	1	µA
		12V		—	2	4	µA
I_{DD}	Operating Current	5V	No load f_{OSC}=150kHz	—	200	400	µA
I_O	Data Output Source Current (D8~D11)	5V	V_{OH}=4.5V	-1	-1.6	—	mA
	Data Output Sink Current (D8~D11)	5V	V_{OL}=0.5V	1	1.6	—	mA
I_{VT}	VT Output Source Current	5V	V_{OH}=4.5V	-1	-1.6	—	mA
	VT Output Sink Current		V_{OL}=0.5V	1	1.6	—	mA
V_{IH}	"H" Input Voltage	5V	—	3.5	—	5	V
V_{IL}	"L" Input Voltage	5V	—	0	—	1	V
f_{OSC}	Oscillator Frequency	5V	R_{OSC}=51kΩ	—	150	—	kHz

Functional Description

Operation

The 2^{12} series of decoders provides various combinations of addresses and data pins in different packages so as to pair with the 2^{12} series of encoders.

The decoders receive data that are transmitted by an encoder and interpret the first N bits of code period as addresses and the last 12–N bits as data, where N is the address code number. A signal on the DIN pin activates the oscillator which in turn decodes the incoming address and data. The decoders will then check the received address three times continuously. If the received address codes all match the contents of the decoder's local address, the 12–N bits of data are decoded to activate the output pins and the VT pin is set high to indicate a valid transmission. This will last unless the address code is incorrect or no signal is received.

The output of the VT pin is high only when the transmission is valid. Otherwise it is always low.

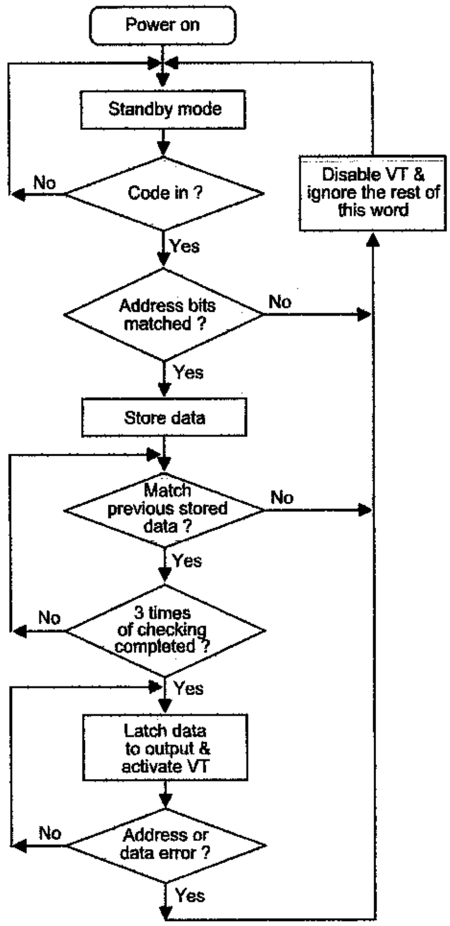
Output type

Of the 2^{12} series of decoders, the HT12F has no data output pin but its VT pin can be used as a momentary data output. The HT12D, on the other hand, provides 4 latch type data pins whose data remain unchanged until new data are received.

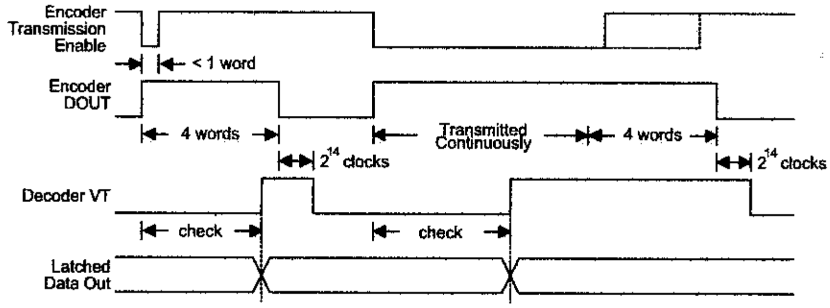
Part No.	Data Pins	Address Pins	Output Type	Operating Voltage
HT12D	4	8	Latch	2.4V~12V
HT12F	0	12	—	2.4V~12V

Flowchart

The oscillator is disabled in the standby state and activated when a logic "high" signal applies to the DIN pin. That is to say, the DIN should be kept low if there is no signal input.



Decoder timing



Encoder/Decoder cross reference table

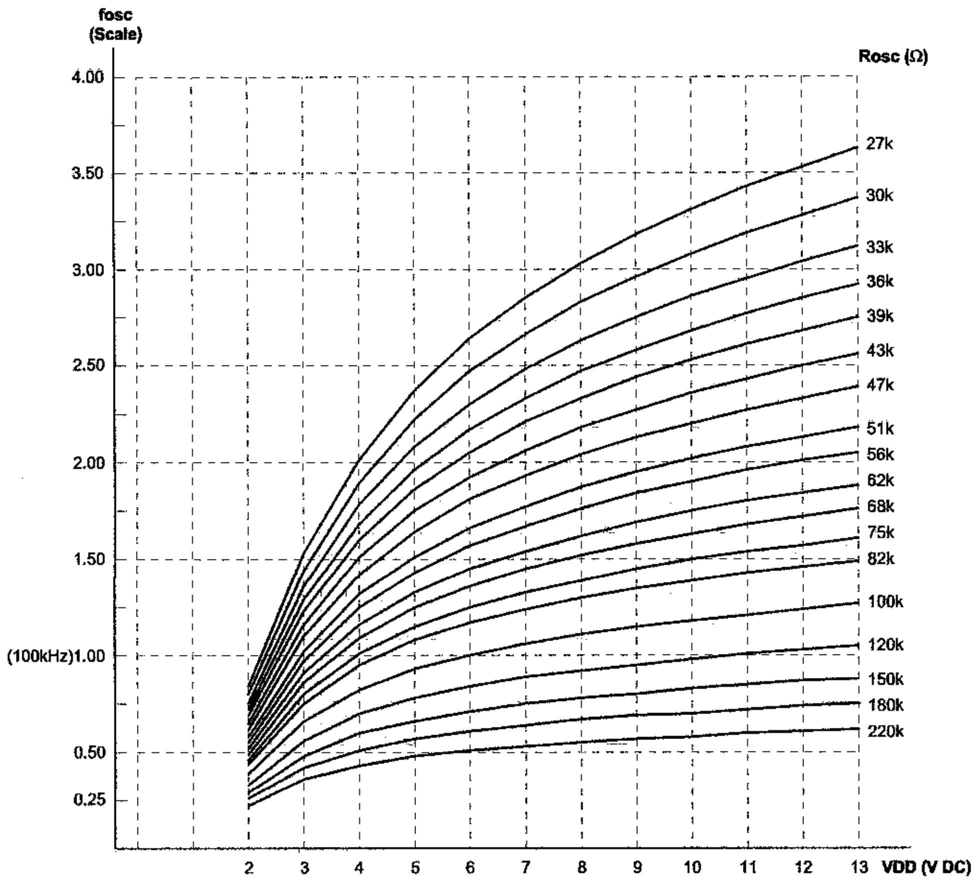
Decoders Part No.	Data Pins	Address Pins	VT	Pair Encoder	Package			
					Encoder		Decoder	
					DIP	SOP	DIP	SOP
HT12D	4	8	√	HT12A	18	20	18	20
				HT12E	18	20		
HT12F	0	12	√	HT12A	18	20	18	20
				HT12E	18	20		

Address/Data sequence

The following table provides address/data sequence for various models of the 2^{12} series of decoders. A correct device should be chosen according to the requirements of the individual addresses and data.

Part No.	Address/Data Bits											
	0	1	2	3	4	5	6	7	8	9	10	11
HT12D	A0	A1	A2	A3	A4	A5	A6	A7	D8	D9	D10	D11
HT12F	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11

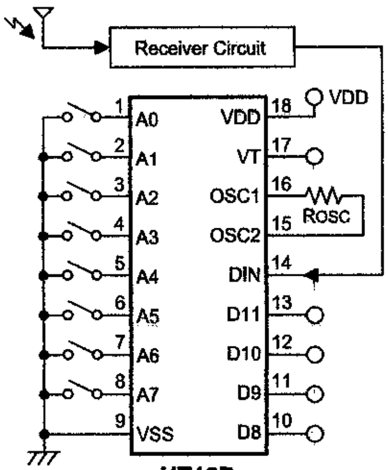
Oscillator frequency vs supply voltage



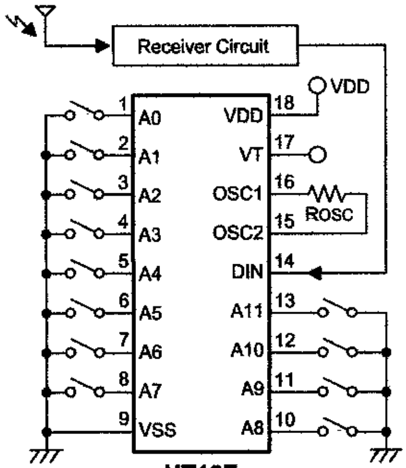
The recommended oscillator frequency is f_{OSCD} (decoder) $\cong$ 50 f_{OSCE} (HT12E encoder)

$\cong \frac{1}{3} f_{OSCE}$ (HT12A encoder).

Application Circuits



HT12D



HT12F

Notes: Typical infrared receiver: PIC-12043T/PIC-12043S (KODESHI CORP.)
or LTM9052 (LITEON CORP.)

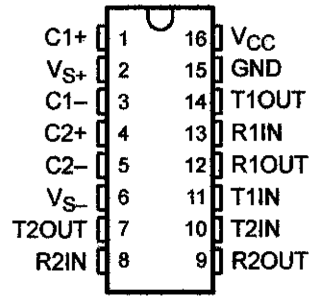
Typical RF receiver: JR-200 (JUWA CORP.)
RE-99 (MING MICROSYSTEM, U.S.A.)

APPENDIX E

MAX232 DATASHEET

- **Meet or Exceed TIA/EIA-232-F and ITU Recommendation V.28**
- **Operate With Single 5-V Power Supply**
- **Operate Up to 120 kbit/s**
- **Two Drivers and Two Receivers**
- **±30-V Input Levels**
- **Low Supply Current . . . 8 mA Typical**
- **Designed to be Interchangeable With Maxim MAX232**
- **ESD Protection Exceeds JESD 22**
  - **2000-V Human-Body Model (A114-A)**
- **Applications**
  - **TIA/EIA-232-F**
  - **Battery-Powered Systems**
  - **Terminals**
  - **Modems**
  - **Computers**

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube	MAX232N	MAX232N
	SOIC (D)	Tube	MAX232D	MAX232
		Tape and reel	MAX232DR	
	SOIC (DW)	Tube	MAX232DW	MAX232
		Tape and reel	MAX232DWR	
	SOP (NS)	Tape and reel	MAX232NSR	MAX232
–40°C to 85°C	PDIP (N)	Tube	MAX232IN	MAX232IN
	SOIC (D)	Tube	MAX232ID	MAX232I
		Tape and reel	MAX232IDR	
	SOIC (DW)	Tube	MAX232IDW	MAX232I
		Tape and reel	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC is a trademark of Texas Instruments.

Function Tables

EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

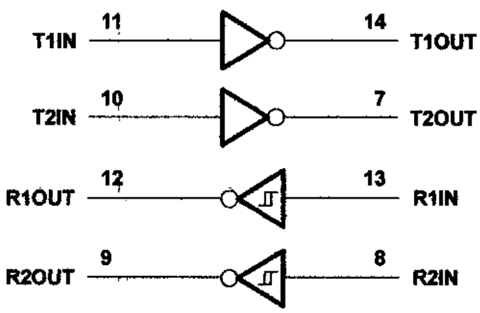
H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Input supply voltage range, V_{CC} (see Note 1) .... –0.3 V to 6 V
Positive output supply voltage range, V_{S+} .... V_{CC} – 0.3 V to 15 V
Negative output supply voltage range, V_{S-} .... –0.3 V to –15 V
Input voltage range, V_I: Driver .... –0.3 V to V_{CC} + 0.3 V
Receiver .... ±30 V
Output voltage range, V_O: T1OUT, T2OUT .... V_{S-} – 0.3 V to V_{S+} + 0.3 V
R1OUT, R2OUT .... –0.3 V to V_{CC} + 0.3 V
Short-circuit duration: T1OUT, T2OUT .... Unlimited
Package thermal impedance, θ_{JA} (see Note 2): D package .... 73°C/W
DW package .... 57°C/W
N package .... 67°C/W
NS package .... 64°C/W
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds .... 260°C
Storage temperature range, T_{stg} .... –65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values are with respect to network ground terminal.
2. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage		4.5	5	5.5	V
V_{IH}	High-level input voltage (T1IN,T2IN)		2			V
V_{IL}	Low-level input voltage (T1IN, T2IN)				0.8	V
R1IN, R2IN	Receiver input voltage				±30	V
T_A	Operating free-air temperature	MAX232	0		70	°C
		MAX232I	–40		85	

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 3 and Figure 4)

PARAMETER		TEST CONDITIONS	MIN	TYP‡	MAX	UNIT
I_{CC}	Supply current	V_{CC} = 5.5 V, All outputs open, T_A = 25°C		8	10	mA

‡ All typical values are at V_{CC} = 5 V and T_A = 25°C.

NOTE 3: Test conditions are C1–C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

DRIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 3)

PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage	T1OUT, T2OUT	R_L = 3 kΩ to GND	5	7		V
V_{OL}	Low-level output voltage‡	T1OUT, T2OUT	R_L = 3 kΩ to GND		–7	–5	V
r_o	Output resistance	T1OUT, T2OUT	$V_{S+} = V_{S-} = 0$, $V_O = \pm 2$ V	300			Ω
I_{OS}§	Short-circuit output current	T1OUT, T2OUT	V_{CC} = 5.5 V, V_O = 0		±10		mA
I_{IS}	Short-circuit input current	T1IN, T2IN	V_I = 0			200	μA

† All typical values are at V_{CC} = 5 V, T_A = 25°C.
‡ The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.
§ Not more than one output should be shorted at a time.
NOTE 3: Test conditions are C1–C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 3)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR	Driver slew rate	R_L = 3 kΩ to 7 kΩ, See Figure 2			30	V/μs
SR(t)	Driver transition region slew rate	See Figure 3		3		V/μs
	Data rate	One TOUT switching		120		kbit/s

NOTE 3: Test conditions are C1–C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

RECEIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 3)

PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage	R1OUT, R2OUT	I_{OH} = –1 mA	3.5			V
V_{OL}	Low-level output voltage‡	R1OUT, R2OUT	I_{OL} = 3.2 mA			0.4	V
V_{IT+}	Receiver positive-going input threshold voltage	R1IN, R2IN	V_{CC} = 5 V, T_A = 25°C		1.7	2.4	V
V_{IT-}	Receiver negative-going input threshold voltage	R1IN, R2IN	V_{CC} = 5 V, T_A = 25°C	0.8	1.2		V
V_{hys}	Input hysteresis voltage	R1IN, R2IN	V_{CC} = 5 V	0.2	0.5	1	V
r_i	Receiver input resistance	R1IN, R2IN	V_{CC} = 5, T_A = 25°C	3	5	7	kΩ

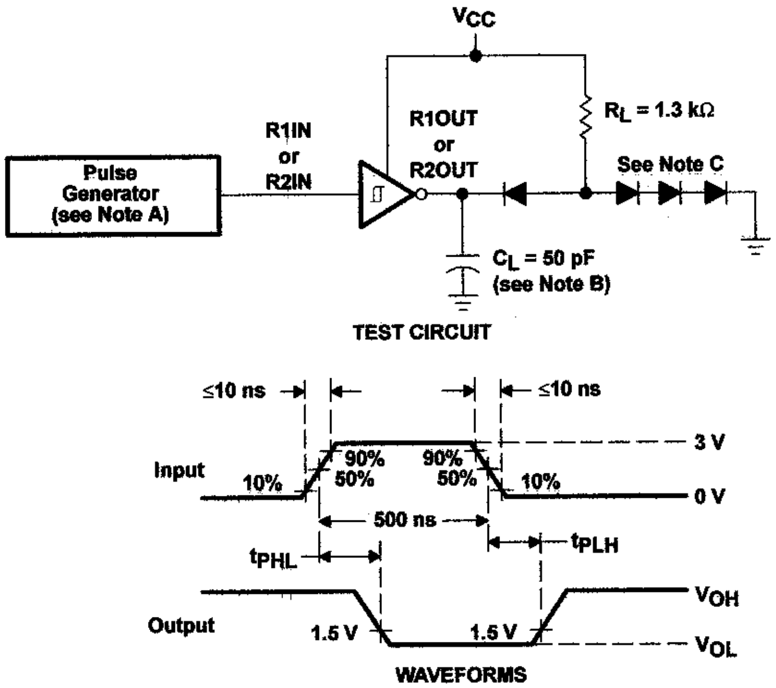
† All typical values are at V_{CC} = 5 V, T_A = 25°C.
‡ The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.
NOTE 3: Test conditions are C1–C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 3 and Figure 1)

PARAMETER		TYP	UNIT
$t_{PLH(R)}$	Receiver propagation delay time, low- to high-level output	500	ns
$t_{PHL(R)}$	Receiver propagation delay time, high- to low-level output	500	ns

NOTE 3: Test conditions are C1–C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

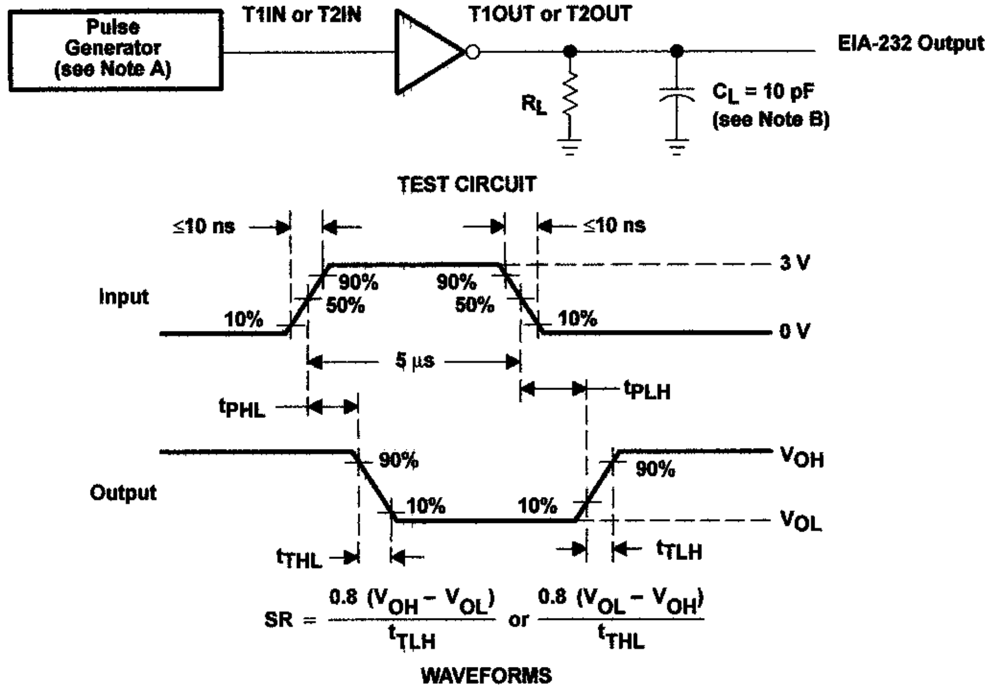
PARAMETER MEASUREMENT INFORMATION



NOTES: A. The pulse generator has the following characteristics: Z_O = 50 Ω, duty cycle ≤ 50%.
B. C_L includes probe and jig capacitance.
C. All diodes are 1N3064 or equivalent.

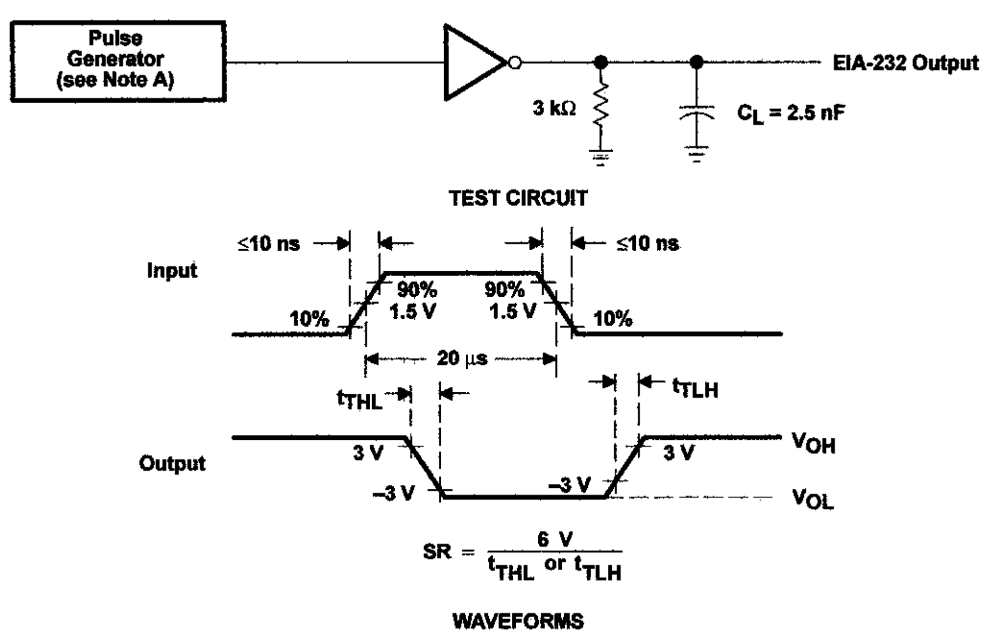
Figure 1. Receiver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements

PARAMETER MEASUREMENT INFORMATION



NOTES: A. The pulse generator has the following characteristics: $Z_O = 50\ \Omega$, duty cycle ≤ 50%.
B. C_L includes probe and jig capacitance.

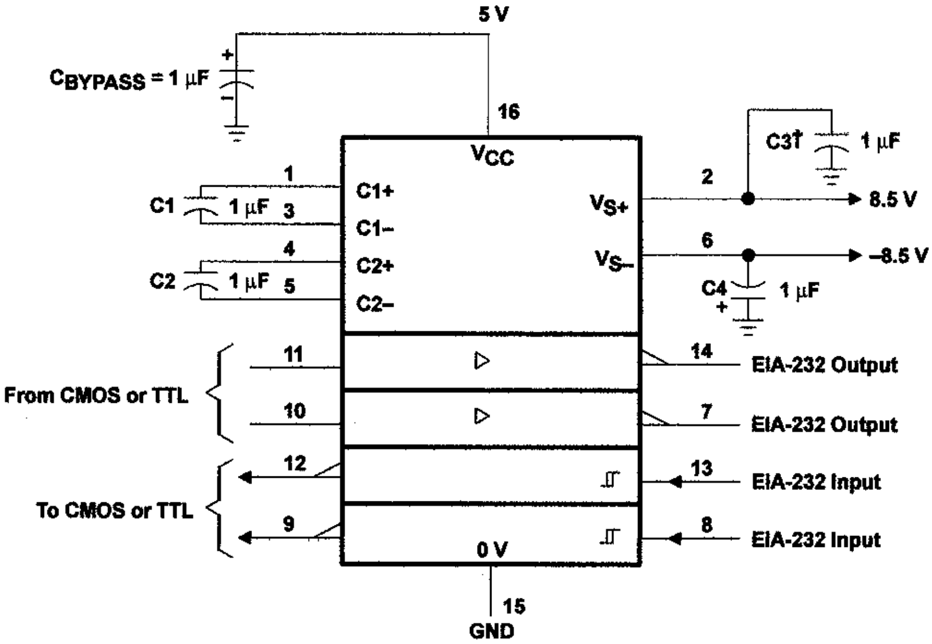
Figure 2. Driver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements (5-µs Input)



NOTE A: The pulse generator has the following characteristics: $Z_O = 50\ \Omega$, duty cycle ≤ 50%.

Figure 3. Test Circuit and Waveforms for t_{THL} and t_{TLH} Measurements (20-µs Input)

APPLICATION INFORMATION



† C3 can be connected to V_{CC} or GND.

Figure 4. Typical Operating Circuit

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265

APPENDIX F

PIC16F84 DATASHEET

MICROCHIP

PIC16F84A

18-pin *Enhanced* FLASH/EEPROM 8-Bit Microcontroller

High Performance RISC CPU Features:

- Only 35 single word instructions to learn
- All instructions single-cycle except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- 1024 words of program memory
- 68 bytes of Data RAM
- 64 bytes of Data EEPROM
- 14-bit wide instruction words
- 8-bit wide data bytes
- 15 Special Function Hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
  - External RB0/INT pin
  - TMR0 timer overflow
  - PORTB<7:4> interrupt-on-change
  - Data EEPROM write complete

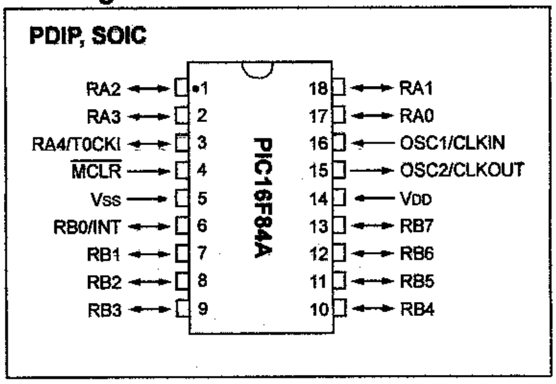
Peripheral Features:

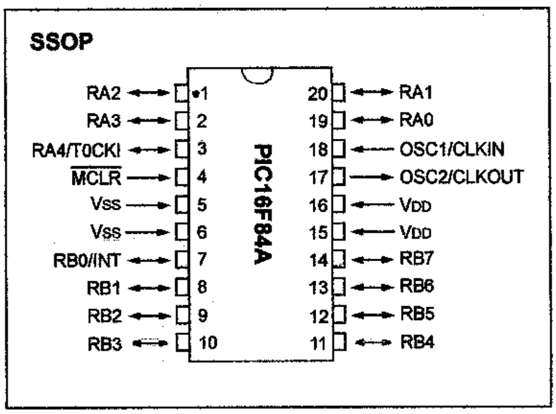
- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
  - 25 mA sink max. per pin
  - 25 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

Special Microcontroller Features:

- 10,000 erase/write cycles *Enhanced* FLASH Program memory typical
- 10,000,000 typical erase/write cycles EEPROM Data memory typical
- EEPROM Data Retention > 40 years
- In-Circuit Serial Programming™ (ICSP™) - via two pins
- Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation
- Code protection
- Power saving SLEEP mode
- Selectable oscillator options

Pin Diagrams





CMOS Enhanced FLASH/EEPROM Technology:

- Low power, high speed technology
- Fully static design
- Wide operating voltage range:
  - Commercial: 2.0V to 5.5V
  - Industrial: 2.0V to 5.5V
- Low power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 15 μA typical @ 2V, 32 kHz
  - < 0.5 μA typical standby current @ 2V

TABLE 1-1: PIC16F84A PINOUT DESCRIPTION

Pin Name	PDIP No.	SOIC No.	SSOP No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	16	18	I	ST/CMOS(3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	15	19	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
$\overline{\text{MCLR}}$	4	4	4	I/P	ST	Master Clear (Reset) input/programming voltage input. This pin is an active low RESET to the device.
						PORTA is a bi-directional I/O port.
RA0	17	17	19	I/O	TTL	
RA1	18	18	20	I/O	TTL	
RA2	1	1	1	I/O	TTL	
RA3	2	2	2	I/O	TTL	
RA4/T0CKI	3	3	3	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open drain type.
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	6	7	I/O	TTL/ST(1)	RB0/INT can also be selected as an external interrupt pin.
RB1	7	7	8	I/O	TTL	
RB2	8	8	9	I/O	TTL	
RB3	9	9	10	I/O	TTL	
RB4	10	10	11	I/O	TTL	Interrupt-on-change pin.
RB5	11	11	12	I/O	TTL	Interrupt-on-change pin.
RB6	12	12	13	I/O	TTL/ST(2)	Interrupt-on-change pin. Serial programming clock.
RB7	13	13	14	I/O	TTL/ST(2)	Interrupt-on-change pin. Serial programming data.
VSS	5	5	5,6	P	—	Ground reference for logic and I/O pins.
VDD	14	14	15,16	P	—	Positive supply for logic and I/O pins.

Legend: I= input O = Output I/O = Input/Output P = Power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

2.2 Data Memory Organization

The data memory is partitioned into two areas. The first is the Special Function Registers (SFR) area, while the second is the General Purpose Registers (GPR) area. The SFRs control the operation of the device.

Portions of data memory are banked. This is for both the SFR area and the GPR area. The GPR area is banked to allow greater than 116 bytes of general purpose RAM. The banked areas of the SFR are for the registers that control the peripheral functions. Banking requires the use of control bits for bank selection. These control bits are located in the STATUS Register. Figure 2-2 shows the data memory map organization.

Instructions `MOVWF` and `MOVF` can move values from the W register to any location in the register file ("F"), and vice-versa.

The entire data memory can be accessed either directly using the absolute address of each register file or indirectly through the File Select Register (FSR) (Section 2.5). Indirect addressing uses the present value of the RP0 bit for access into the banked areas of data memory.

Data memory is partitioned into two banks which contain the general purpose registers and the special function registers. Bank 0 is selected by clearing the RP0 bit (STATUS<5>). Setting the RP0 bit selects Bank 1. Each Bank extends up to 7Fh (128 bytes). The first twelve locations of each Bank are reserved for the Special Function Registers. The remainder are General Purpose Registers, implemented as static RAM.

2.2.1 GENERAL PURPOSE REGISTER FILE

Each General Purpose Register (GPR) is 8-bits wide and is accessed either directly or indirectly through the FSR (Section 2.5).

The GPR addresses in Bank 1 are mapped to addresses in Bank 0. As an example, addressing location 0Ch or 8Ch will access the same GPR.

FIGURE 2-2: REGISTER FILE MAP - PIC16F84A

File Address	Bank 0	Bank 1	File Address
00h	Indirect addr.(1)	Indirect addr.(1)	80h
01h	TMR0	OPTION_REG	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	—	—	87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2(1)	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch–4Fh	68 General Purpose Registers (SRAM)	Mapped (accesses) in Bank 0	8Ch–CFh
50h–7Fh			D0h–FFh

☐ Unimplemented data memory location, read as '0'.

Note 1: Not a physical register.

2.3 Special Function Registers

The Special Function Registers (Figure 2-2 and Table 2-1) are used by the CPU and Peripheral functions to control the device operation. These registers are static RAM.

The special function registers can be classified into two sets, core and peripheral. Those associated with the core functions are described in this section. Those related to the operation of the peripheral features are described in the section for that specific feature.

TABLE 2-1: SPECIAL FUNCTION REGISTER FILE SUMMARY

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on RESET	Details on page
Bank 0											
00h	INDF	Uses contents of FSR to address Data Memory (not a physical register)								---- ----	11
01h	TMR0	8-bit Real-Time Clock/Counter								xxxx xxxx	20
02h	PCL	Low Order 8 bits of the Program Counter (PC)								0000 0000	11
03h	STATUS[2]	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	8
04h	FSR	Indirect Data Memory Address Pointer 0								xxxx xxxx	11
05h	PORTA[4]	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	16
06h	PORTB[5]	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	18
07h	—	Unimplemented location, read as '0'								—	—
08h	EEDATA	EEPROM Data Register								xxxx xxxx	13,14
09h	EEADR	EEPROM Address Register								xxxx xxxx	13,14
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of the PC[1]					---0 0000	11
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	10
Bank 1											
80h	INDF	Uses Contents of FSR to address Data Memory (not a physical register)								---- ----	11
81h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	9
82h	PCL	Low order 8 bits of Program Counter (PC)								0000 0000	11
83h	STATUS [2]	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	8
84h	FSR	Indirect data memory address pointer 0								xxxx xxxx	11
85h	TRISA	—	—	—	PORTA Data Direction Register					---1 1111	16
86h	TRISB	PORTB Data Direction Register								1111 1111	18
87h	—	Unimplemented location, read as '0'								—	—
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	13
89h	EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	14
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC[1]					---0 0000	11
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	10

Legend: x = unknown, u = unchanged. - = unimplemented, read as '0', q = value depends on condition

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a slave register for PC<12:8>. The contents of PCLATH can be transferred to the upper byte of the program counter, but the contents of PC<12:8> are never transferred to PCLATH.

2: The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ status bits in the STATUS register are not affected by a $\overline{\text{MCLR}}$ Reset.

3: Other (non power-up) RESETS include: external RESET through $\overline{\text{MCLR}}$ and the Watchdog Timer Reset.

4: On any device RESET, these pins are configured as inputs.

5: This is the value that will be in the port output latch.

APPENDIX G

PIC16F877 DATASHEET

PIC16F87XA

28/40/44-Pin Enhanced Flash Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input
  DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I^2C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (V_{REF}) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

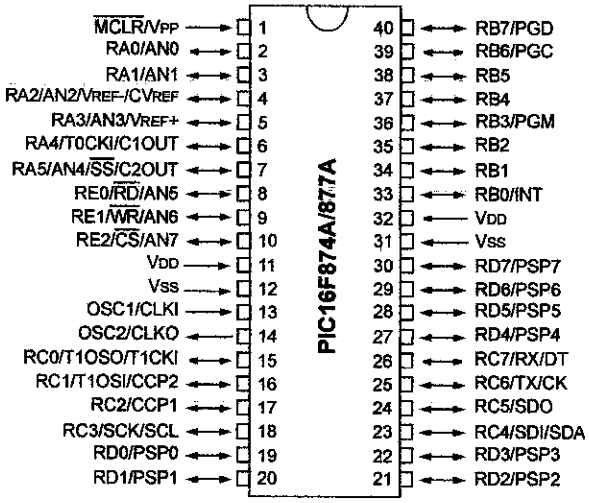
CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

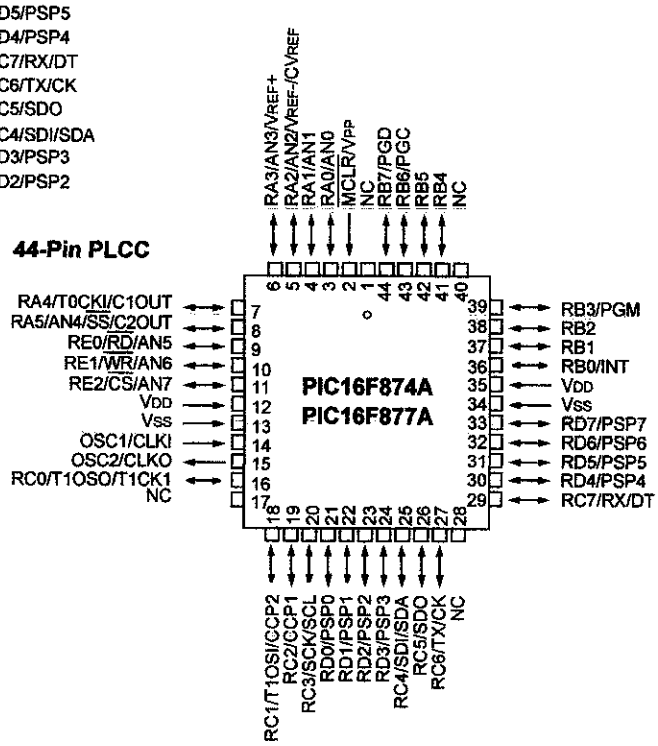
Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I^2C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

Pin Diagrams (Continued)

40-Pin PDIP



44-Pin PLCC



44-Pin TQFP

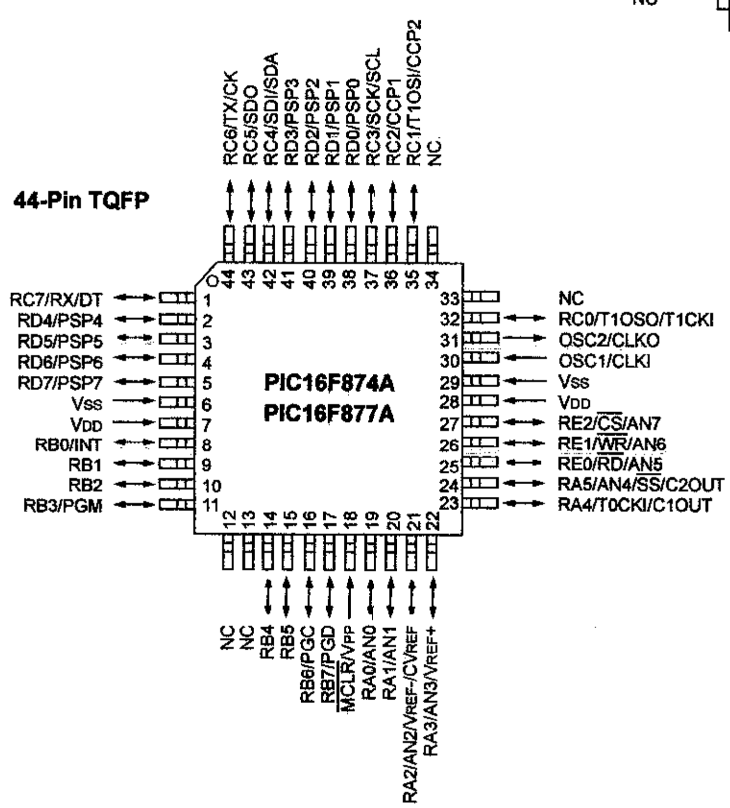


TABLE 1-2: PIC16F873A/876A PINOUT DESCRIPTION

Pin Name	PDIP, SOIC, SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI	9	6		ST/CMOS(3)	Oscillator crystal or external clock input.
OSC1			I		Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS.
CLKI			I		External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO	10	7		—	Oscillator crystal or clock output.
OSC2			O		Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO			O		In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	1	26		ST	Master Clear (input) or programming voltage (output).
$\overline{MCLR}$			I		Master Clear (Reset) input. This pin is an active low Reset to the device.
VPP			P		Programming voltage input.
					PORTA is a bidirectional I/O port.
RA0/AN0	2	27		TTL	
RA0			I/O		Digital I/O.
AN0			I		Analog input 0.
RA1/AN1	3	28		TTL	
RA1			I/O		Digital I/O.
AN1			I		Analog input 1.
RA2/AN2/VREF-/CVREF	4	1		TTL	
RA2			I/O		Digital I/O.
AN2			I		Analog input 2.
VREF-			I		A/D reference voltage (Low) input.
CVREF			O		Comparator VREF output.
RA3/AN3/VREF+	5	2		TTL	
RA3			I/O		Digital I/O.
AN3			I		Analog input 3.
VREF+			I		A/D reference voltage (High) input.
RA4/T0CKI/C1OUT	6	3		ST	
RA4			I/O		Digital I/O – Open-drain when configured as output.
T0CKI			I		Timer0 external clock input.
C1OUT			O		Comparator 1 output.
RA5/AN4/$\overline{SS}$/C2OUT	7	4		TTL	
RA5			I/O		Digital I/O.
AN4			I		Analog input 4.
$\overline{SS}$			I		SPI slave select input.
C2OUT			O		Comparator 2 output.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

TABLE 1-2: PIC16F873A/876A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP, SOIC, SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
					PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/INT	21	18		TTL/ST(1)	
RB0			I/O		Digital I/O.
INT			I		External interrupt.
RB1	22	19	I/O	TTL	Digital I/O.
RB2	23	20	I/O	TTL	Digital I/O.
RB3/PGM	24	21		TTL	
RB3			I/O		Digital I/O.
PGM			I		Low-voltage (single-supply) ICSP programming enable pin.
RB4	25	22	I/O	TTL	Digital I/O.
RB5	26	23	I/O	TTL	Digital I/O.
RB6/PGC	27	24		TTL/ST(2)	
RB6			I/O		Digital I/O.
PGC			I		In-circuit debugger and ICSP programming clock.
RB7/PGD	28	25		TTL/ST(2)	
RB7			I/O		Digital I/O.
PGD			I/O		In-circuit debugger and ICSP programming data.
					PORTC is a bidirectional I/O port.
RC0/T1OSO/T1CKI	11	8		ST	
RC0			I/O		Digital I/O.
T1OSO			O		Timer1 oscillator output.
T1CKI			I		Timer1 external clock input.
RC1/T1OSI/CCP2	12	9		ST	
RC1			I/O		Digital I/O.
T1OSI			I		Timer1 oscillator input.
CCP2			I/O		Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1	13	10		ST	
RC2			I/O		Digital I/O.
CCP1			I/O		Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL	14	11		ST	
RC3			I/O		Digital I/O.
SCK			I/O		Synchronous serial clock input/output for SPI mode.
SCL			I/O		Synchronous serial clock input/output for I²C mode.
RC4/SDI/SDA	15	12		ST	
RC4			I/O		Digital I/O.
SDI			I		SPI data in.
SDA			I/O		I²C data I/O.
RC5/SDO	16	13		ST	
RC5			I/O		Digital I/O.
SDO			O		SPI data out.
RC6/TX/CK	17	14		ST	
RC6			I/O		Digital I/O.
TX			O		USART asynchronous transmit.
CK			I/O		USART1 synchronous clock.
RC7/RX/DT	18	15		ST	
RC7			I/O		Digital I/O.
RX			I		USART asynchronous receive.
DT			I/O		USART synchronous data.
V_{SS}	8, 19	5, 6	P	—	Ground reference for logic and I/O pins.
V_{DD}	20	17	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI	13	14	30	32		ST/CMOS(4)	Oscillator crystal or external clock input.
OSC1					I		Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS.
CLKI					I		External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO	14	15	31	33		—	Oscillator crystal or clock output.
OSC2					O		Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO					O		In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	1	2	18	18		ST	Master Clear (input) or programming voltage (output).
$\overline{MCLR}$					I		Master Clear (Reset) input. This pin is an active low Reset to the device.
VPP					P		Programming voltage input.
							PORTA is a bidirectional I/O port.
RA0/AN0	2	3	19	19		TTL	
RA0					I/O		Digital I/O.
AN0					I		Analog input 0.
RA1/AN1	3	4	20	20		TTL	
RA1					I/O		Digital I/O.
AN1					I		Analog input 1.
RA2/AN2/VREF-/CVREF	4	5	21	21		TTL	
RA2					I/O		Digital I/O.
AN2					I		Analog input 2.
VREF-					I		A/D reference voltage (Low) input.
CVREF					O		Comparator VREF output.
RA3/AN3/VREF+	5	6	22	22		TTL	
RA3					I/O		Digital I/O.
AN3					I		Analog input 3.
VREF+					I		A/D reference voltage (High) input.
RA4/T0CKI/C1OUT	6	7	23	23		ST	
RA4					I/O		Digital I/O – Open-drain when configured as output.
T0CKI					I		Timer0 external clock input.
C1OUT					O		Comparator 1 output.
RA5/AN4/$\overline{SS}$/C2OUT	7	8	24	24		TTL	
RA5					I/O		Digital I/O.
AN4					I		Analog input 4.
$\overline{SS}$					I		SPI slave select input.
C2OUT					O		Comparator 2 output.

Legend: I = input   O = output   I/O = input/output   P = power
— = Not used   TTL = TTL input   ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
							PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	33	36	8	9		TTL/ST(1)	
RB0					I/O		Digital I/O.
INT					I		External interrupt.
RB1	34	37	9	10	I/O	TTL	Digital I/O.
RB2	35	38	10	11	I/O	TTL	Digital I/O.
RB3/PGM	36	39	11	12		TTL	
RB3					I/O		Digital I/O.
PGM					I		Low-voltage ICSP programming enable pin.
RB4	37	41	14	14	I/O	TTL	Digital I/O.
RB5	38	42	15	15	I/O	TTL	Digital I/O.
RB6/PGC	39	43	16	16		TTL/ST(2)	
RB6					I/O		Digital I/O.
PGC					I		In-circuit debugger and ICSP programming clock.
RB7/PGD	40	44	17	17		TTL/ST(2)	
RB7					I/O		Digital I/O.
PGD					I/O		In-circuit debugger and ICSP programming data.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
							PORTC is a bidirectional I/O port.
RC0/T1OSO/T1CKI	15	16	32	34		ST	
RC0					I/O		Digital I/O.
T1OSO					O		Timer1 oscillator output.
T1CKI					I		Timer1 external clock input.
RC1/T1OSI/CCP2	16	18	35	35		ST	
RC1					I/O		Digital I/O.
T1OSI					I		Timer1 oscillator input.
CCP2					I/O		Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1	17	19	36	36		ST	
RC2					I/O		Digital I/O.
CCP1					I/O		Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL	18	20	37	37		ST	
RC3					I/O		Digital I/O.
SCK					I/O		Synchronous serial clock input/output for SPI mode.
SCL					I/O		Synchronous serial clock input/output for I^{2}C mode.
RC4/SDI/SDA	23	25	42	42		ST	
RC4					I/O		Digital I/O.
SDI					I		SPI data in.
SDA					I/O		I^{2}C data I/O.
RC5/SDO	24	26	43	43		ST	
RC5					I/O		Digital I/O.
SDO					O		SPI data out.
RC6/TX/CK	25	27	44	44		ST	
RC6					I/O		Digital I/O.
TX					O		USART asynchronous transmit.
CK					I/O		USART1 synchronous clock.
RC7/RX/DT	26	29	1	1		ST	
RC7					I/O		Digital I/O.
RX					I		USART asynchronous receive.
DT					I/O		USART synchronous data.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

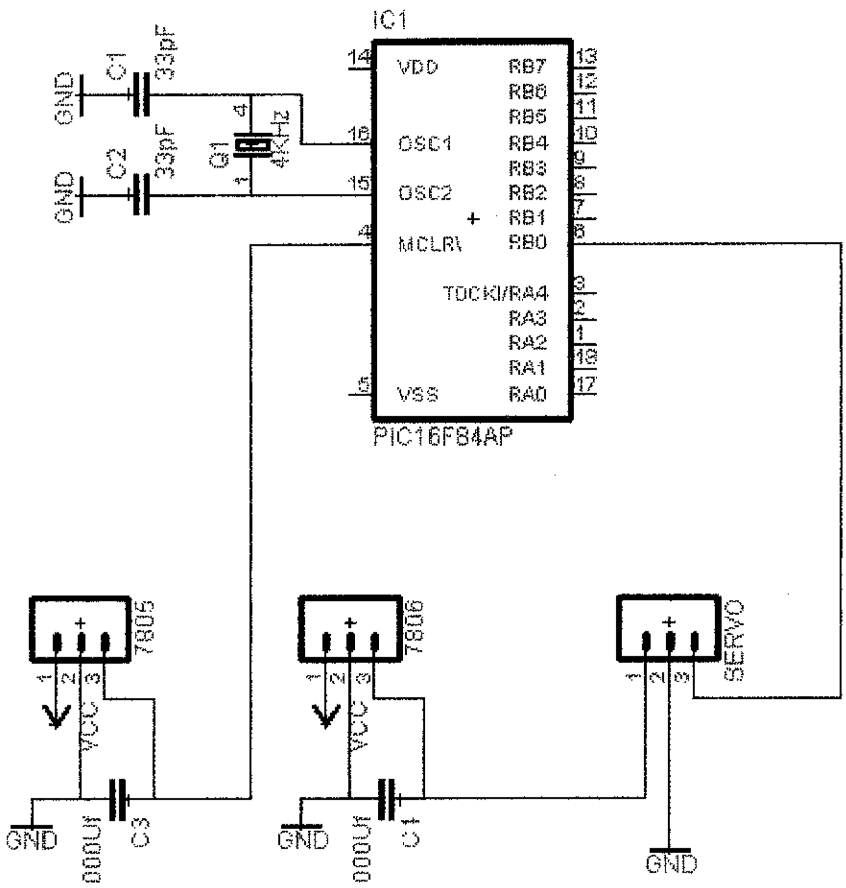
TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
							PORTD is a bidirectional I/O port or Parallel Slave Port when interfacing to a microprocessor bus.
RD0/PSP0	19	21	38	38		ST/TTL(3)	
RD0					I/O		Digital I/O.
PSP0					I/O		Parallel Slave Port data.
RD1/PSP1	20	22	39	39		ST/TTL(3)	
RD1					I/O		Digital I/O.
PSP1					I/O		Parallel Slave Port data.
RD2/PSP2	21	23	40	40		ST/TTL(3)	
RD2					I/O		Digital I/O.
PSP2					I/O		Parallel Slave Port data.
RD3/PSP3	22	24	41	41		ST/TTL(3)	
RD3					I/O		Digital I/O.
PSP3					I/O		Parallel Slave Port data.
RD4/PSP4	27	30	2	2		ST/TTL(3)	
RD4					I/O		Digital I/O.
PSP4					I/O		Parallel Slave Port data.
RD5/PSP5	28	31	3	3		ST/TTL(3)	
RD5					I/O		Digital I/O.
PSP5					I/O		Parallel Slave Port data.
RD6/PSP6	29	32	4	4		ST/TTL(3)	
RD6					I/O		Digital I/O.
PSP6					I/O		Parallel Slave Port data.
RD7/PSP7	30	33	5	5		ST/TTL(3)	
RD7					I/O		Digital I/O.
PSP7					I/O		Parallel Slave Port data.
							PORTE is a bidirectional I/O port.
RE0/$\overline{RD}$/AN5	8	9	25	25		ST/TTL(3)	
RE0					I/O		Digital I/O.
$\overline{RD}$					I		Read control for Parallel Slave Port.
AN5					I		Analog input 5.
RE1/$\overline{WR}$/AN6	9	10	26	26		ST/TTL(3)	
RE1					I/O		Digital I/O.
$\overline{WR}$					I		Write control for Parallel Slave Port.
AN6					I		Analog input 6.
RE2/$\overline{CS}$/AN7	10	11	27	27		ST/TTL(3)	
RE2					I/O		Digital I/O.
$\overline{CS}$					I		Chip select control for Parallel Slave Port.
AN7					I		Analog input 7.
VSS	12, 31	13, 34	6, 29	6, 30, 31	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	12, 35	7, 28	7, 8, 28, 29	P	—	Positive supply for logic and I/O pins.
NC	—	1, 17, 28, 40	12,13, 33, 34	13	—	—	These pins are not internally connected. These pins should be left unconnected.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

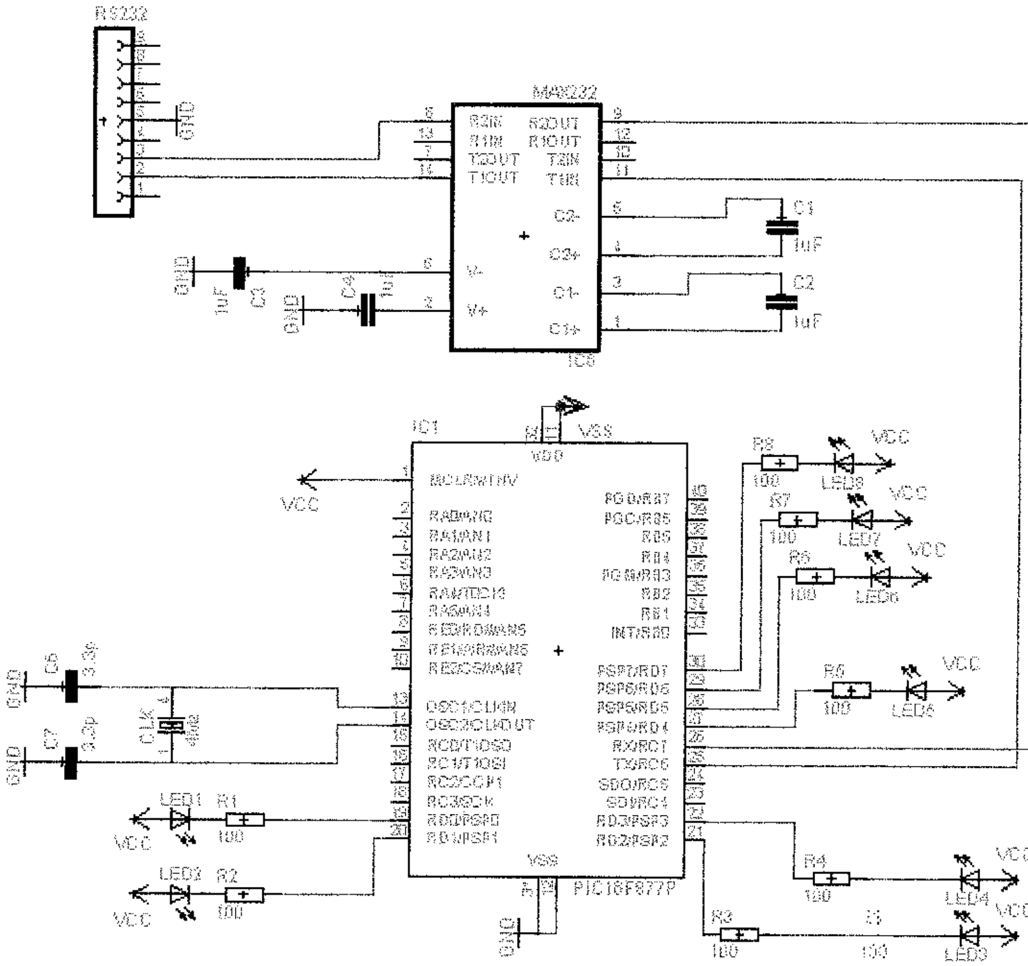
Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

APPENDIX H
SERVO MOTOR CIRCUIT



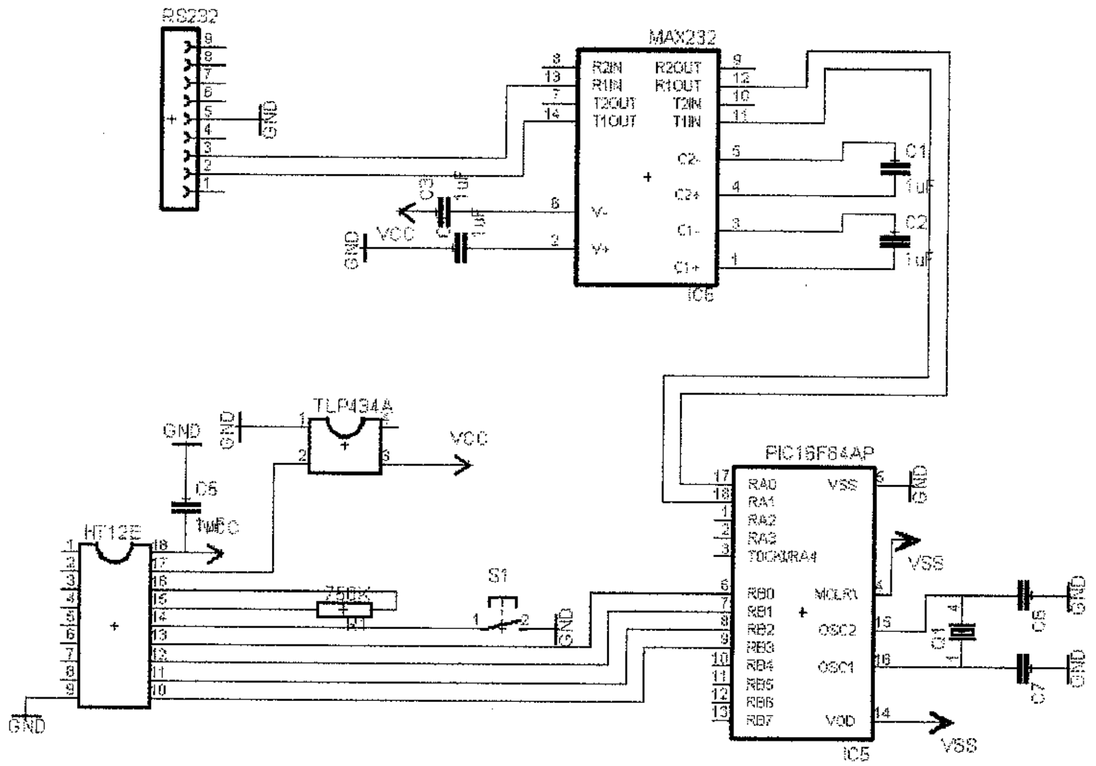
APPENDIX I

CIRCUIT FOR CONNECTING PC TO MICROCONTROLLER

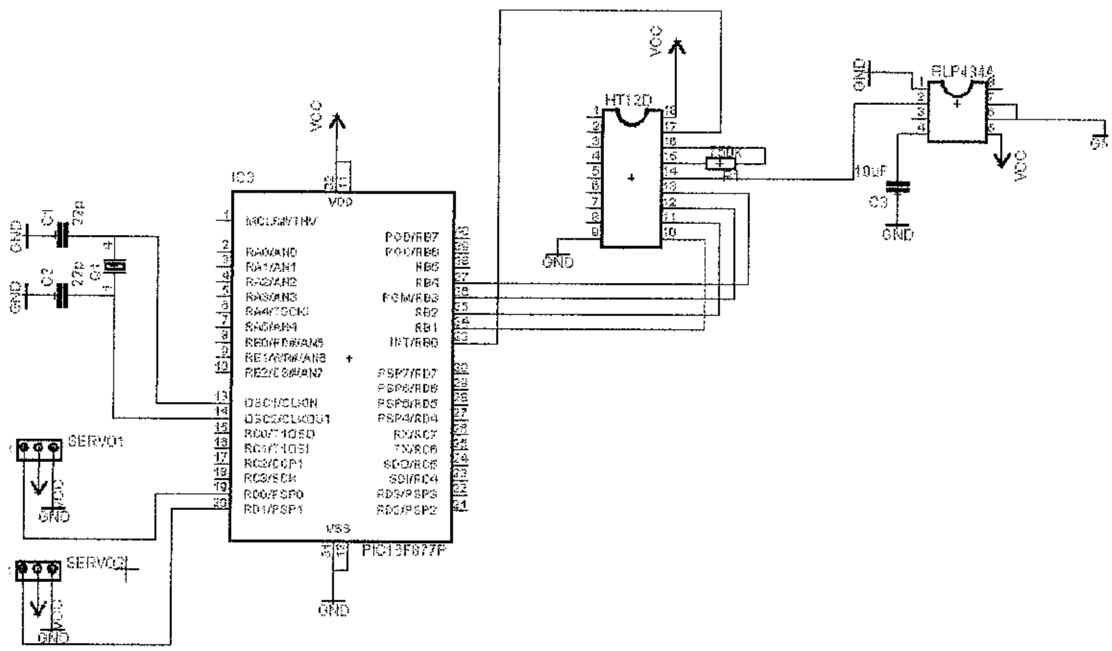


APPENDIX J

TRANSMITTER CIRCUIT

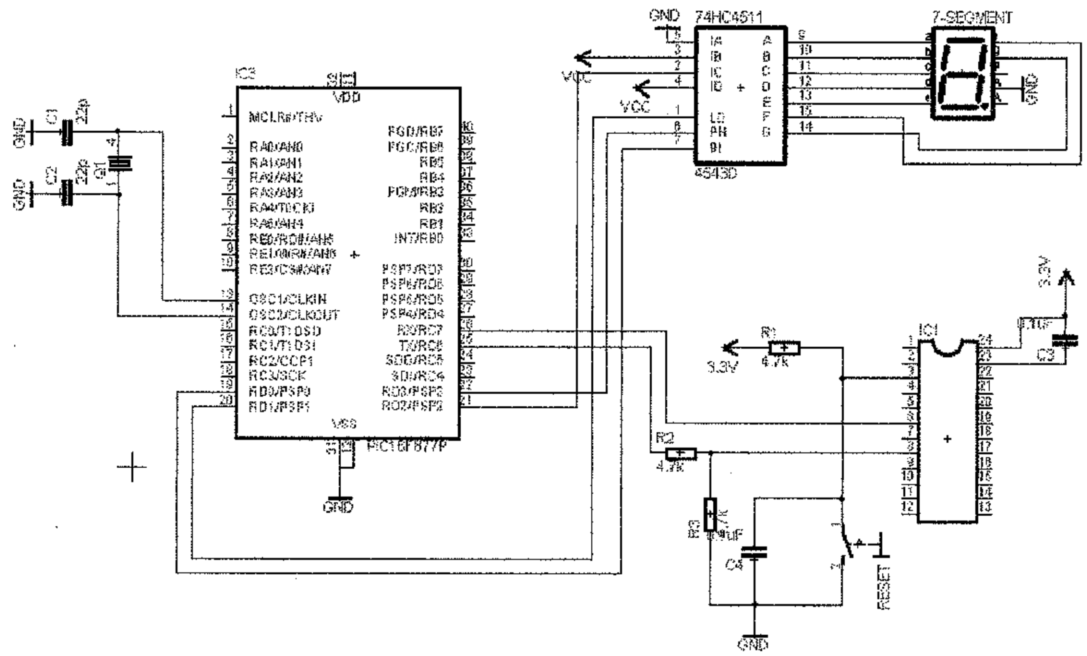


APPENDIX K
RECEIVER CIRCUIT



APPENDIX L

BLUETOOTH CIRCUIT CONNECTION



APPENDIX M

CODING FOR TESTING SERVO MOVEMENT

```
#include <16F84A.h>
#fuses XT,NOWDT,NOPUT,NOPROTECT
#use DELAY(CLOCK=4000000)
void main()
{
 while(1)
   {
    if (input(PIN_A0)==0 && input(PIN_A1)==0 && input(PIN_A2)==0 )
       //no movement
       {
      output_high(PIN_B0);
      delay_us(1500);
      output_low(PIN_B0);
      delay_ms(20);
      }
       else if(input(PIN_A0)==0 && input(PIN_A1)==0 && input(PIN_A2)==1)
      //90 degree cw
      {
      output_high(PIN_B0);
      delay_us(750);
      output_low(PIN_B0);
      delay_ms(20);
      {
      else if(input(PIN_A0)==0 && input(PIN_A1)==1 && input(PIN_A2)==0)
      //45 degree cw
```

```
        {
        output_high(PIN_B0);
        delay_us(1125);
        output_low(PIN_B0);
        delay_ms(20);
        {
        else if(input(PIN_A0)==0 && input(PIN_A1)==1 && input(PIN_A2)==1)
        //90 degree ccw
        {
        //servo on (90 degrees cw turn)
        output_high(PIN_B0);
        delay_us(1875);
        output_low(PIN_B0);
        delay_ms(20);
        {
        else if(input(PIN_A0)==0 && input(PIN_A1)==1 && input(PIN_A2)==1)
        //45 degree ccw
        {
        output_high(PIN_B0);
        delay_us(2250);
        output_low(PIN_B0);
        delay_ms(20);
        }
      else {}
   }
}
```

APPENDIX N

CODING FOR DATA TRANSMISSION FROM PC TO PIC

```
#include <16F877.h>
#fuses XT,NOWDT,NOPUT,NOPROTECT
#use DELAY(CLOCK=4000000)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7, bits=8)
#include <stdio.h>
#include <stdlib.h>
void main(void)
{
   char con;
   // Output pin if first initialized to zero
   output_low(PIN_D0);
   output_low(PIN_D1);
   output_low(PIN_D2);
   output_low(PIN_D3);
   output_low(PIN_D4);
   output_low(PIN_D5);
   output_low(PIN_D6);
   output_low(PIN_D7);
while(1)
  {
      con=getch();
      //wait for and get serial character
      //function only executed when a character is received
      if (con=='w')
      output_high(PIN_D0);
       if (con=='q')
      output_low(PIN_D0);
       if (con=='s')
      output_high(PIN_D1);
       if (con=='e')
```

```
        output_low(PIN_D1);
        if (con=='a')
        output_high(PIN_D2);
        if (con=='z')
        output_low(PIN_D2);
        if (con=='d')
        output_high(PIN_D3);
        if (con=='x')
        output_low(PIN_D3);
    }
}
```

APPENDIX O
CODING FOR TRANSMITTER

```
#include <16F84.h>

#include<stdlib.h>

#include<stdio.h>

#fuses XT,NOWDT,NOPUT,NOPROTECT

#use DELAY(CLOCK=4000000)

#use rs232(baud=9600, xmit=PIN_A0, rcv=PIN_A1, bits=8)

void main()

{

        Set_tris_b(0x00);          //set port B as output port

        char con;

        while(1)

        {

        con = getch();
```

//wait for and get serial character. The function only executed when a character is received

```
        if (con=='w')

        output_low(PIN_B0);

        output_low(PIN_B1);

        output_low(PIN_B2);

        output_high(PIN_B3);
```

//from the function above, if the character 'w' is type, the PIN_B0, PIN_B1 and PIN_B2 in the microcontroller will be in initiated to low and PIN_B3 to high.

```
        if (con=='e')

        output_low(PIN_B0);

        output_low(PIN_B1);

        output_high(PIN_B2);
```

output_low(PIN_B3);

//from the function above, if the character 'e' is type, the PIN_B0, PIN_B1 and PIN_B3 in the microcontroller will be in initiated to low and PIN_B2 to high

if (con=='r')

output_low(PIN_B0);

output_low(PIN_B1);

output_high(PIN_B2);

output_high(PIN_B3);

//from the function above, if the character 'e' is type, the PIN_B0 and PIN_B1 in the microcontroller will be in initiated to low and PIN_B2 and PIN_B3 to high

}

Output_high(PIN_B4);

//The values of data at the output PORT B will be sent out via transmitter but first will be encoded. The data are transmitted together with the header bits via RF transmission medium upon receipt of a trigger signal which is from PIN_B4. The header bit informed the transmitter that information is about to be transmitted.

}

APPENDIX P
CODING FOR RECEIVER

```
#include <16F877.h>
#fuses XT,NOWDT,NOPUT,NOPROTECT
#use DELAY(CLOCK=4000000)
#include<stdio.h>
#include<stdlib.h>
void main()
{
while(1)
{
Data=0;
Data=input_b;
if (input(PIN_B0==0 && input(PIN_B1)==0 && input(PIN_B2)==0 &&
input(PIN_B3)==1)
        {
        output_high(PIN_D0);
        delay_us(1500);
        output_low(PIN_D0);
        delay_ms(20);
        }
//servo at the default position

if (input(PIN_B0==0 && input(PIN_B1)==0 && input(PIN_B2)==1 &&
input(PIN_B3)==0)
        {
         output_high(PIN_D0);
         delay_us(2250);
         output_low(PIN_D0);
         delay_ms(20);
        }
//servo turn 90 degree CW
```

```
if (input(PIN_B0==0 && input(PIN_B1)==0 && input(PIN_B2)==1 &&
input(PIN_B3)==1)
        {
         output_high(PIN_D0);
         delay_us(750);
         output_low(PIN_D0);
         delay_ms(20);
        }
//servo 90 degree CCW
else {}
}
}
```

APPENDIX Q
CODING FOR TESTING OF BLUETOOTH DATA TRANSMISSION

```
include <pic.h>
__CONFIG(0x3F32);
#define seg PORTD                       // define 7 segment as PORTD
unsigned char a;
void init(void)                         // subroutine to initialize
{
        SPBRG=0x0A;                     // set baud rate as 115200 baud
        BRGH=1;
        TXEN=1;
        CREN=1;
        SPEN=1;
        TRISD = 0b00000000;
        seg   = 0b00000000;
}
void display(unsigned char c)           // subroutine to display the text on the screen
{
        while (TXIF == 0);
        TXREG = c;
}
unsigned char receive(void)             // subroutine to receive text from PC
{
        while (RCIF == 0);
        a = RCREG;
        return a;
}
void main(void)
{
        init();
        while(1)                        // wait for 'ok' to be entered
```

```
{
        a = receive();
        if (a == 'o')
        {
        a = receive();
        if (a == 'k') break;
        }
}
        display('P');
        display('r');
        display('e');
        display('s');
        display('s');
        display(0x20);
        display('a');
        display('n');
        display('y');
        display(0x20);
        display('n');
        display('u');
        display('m');
        display('b');
        display('e');
        display('r');
        seg = 1;
        while(1)        // wait for number and display it on 7 segment
        {       a = receive();
                if
(a=='1'||a=='2'||a=='3'||a=='4'||a=='5'||a=='6'||a=='7'||a=='8'||a=='9'||a=='0')
                {
                seg = a-0x30;
                }
        }
}
```

APPENDIX R

CODING FOR WIRELESS PANT AND TILT SURVEILLANCE PLATFORM (BLUETOOTH)

```
#include <pic.h>

__CONFIG(0x3F32);

unsigned char a;

void init(void)                 // subroutine to initialize
{
        SPBRG=0x0A;             // set baud rate as 115200 baud
        BRGH=1;
        TXEN=1;
        CREN=1;
        SPEN=1;
        TRISD = 0b00000000;     //PORTD as output port

}

void display(unsigned char c)   // subrountine to display the text on the screen

{
        while (TXIF == 0);
        TXREG = c;
}

unsigned char receive(void)     // subrountine to receive text from PC
{
        while (RCIF == 0);
        a = RCREG;
        return a;
```

```
}

void main(void)
{
        init();

        while(1)                        // wait for 'ok' to be entered
        {
                a = receive();
                if (a == 'o')
                        {
                                a = receive();
                                if (a == 'k') break;
                        }
        }


                display('C');           // change the text for different display
                display('y');
                display('t');
                display('r');
                display('o');
                display('n');
                display(0x0a);
                display(0x0d);
                display('P');
                display('r');
                display('e');
                display('s');
                display('s');
                display(0x20);
                display('a');
                display('n');
                display('y');
```

```
display(0x20);
display('n');
display('u');
display('m');
display('b');
display('e');
display('r');




while (1)
{
a = receive();

if (a == '1')
{PORTD = 0b00000001;}            //pan 90 degree left (PanCam1)

else if (a == '2')
{PORTD = 0b00000001;}            //pan 45 degree left (PanCam1)

else if (a == '3')
{PORTD = 0b00000001;}            //0 degree (PanCam1)

else if (a == '4')
{PORTD = 0b00000100;}            //pan 45 degree right (PanCam1)

else if (a == '5')
{PORTD = 0b00000101;}            //pan 90 degree right (PanCam1)

else if (a == '6')
{PORTD = 0b00000110;}            //tilt 45 degree up (TiltCam1)

else if (a == '7')
{PORTD = 0b00000111;}            //0 degree (TiltCam1)
```

```
        else if (a == '8')
        {PORTD = 0b00001000;}          //tilt 45 degree down (TiltCam1)

        else if (a == '9')
        {PORTD = 0b00001001;}          //pan 90 degree left (PanCam2)

        else if (a == '0')
        {PORTD = 0b00001010;}          //pan 45 degree left (PanCam2)

        else if (a == 'q')
        {PORTD = 0b00001011;}          //0 degree (PanCam2)

        else if (a == 'w')
        {PORTD = 0b00001100;}          //pan 45 degree right (PanCam2)

        else if (a == 'e')
        {PORTD = 0b00001101;}          //pan 90 degree right (PanCam2)

        else if (a == 'r')
        {PORTD = 0b00001110;}          //tilt 45 degree up (TiltCam2)

        else if (a == 't')
        {PORTD = 0b00001111;}          //0 degree (TiltCam2)

        else if (a == 'y')
        {PORTD = 0b00010000;}          //tilt 45 degree down (TiltCam2)

        else
        {PORTD = 0b00000000;}          //motor stop
        }
}
```

APPENDIX S

CODING FOR WIRELESS PANT AND TILT SURVEILLANCE PLATFORM (SERVO)

```
/*
  Pin Declaration
  -------------------
  PIN_RB0 = output to the PanCam1
  PIN_RB1 = output to the TiltCam1
  PIN_RB2 = output to the PanCam2
  PIN_RB3 = output to the TiltCam2
*/
#include <16F877.h>
#fuses XT,NOWDT,NOPUT,NOPROTECT
#use DELAY(CLOCK=4000000)
#include<stdio.h>
void main(void)
{
   output_low(PIN_B0);      //output to PanCam1
   output_low(PIN_B1);      //output to TiltCam1
   output_low(PIN_B2);      //output to PanCam2
   output_low(PIN_B3);      //output to TilCam2
   while(1)
   {
       if (input(PIN_D4)==0 && input(PIN_D3)==0 && input(PIN_D2)==0 &&
       input(PIN_D1)==0 && input(PIN_D0)==1)
       // input=00001 90 degree left (PanCam1)
       {
              output_high(PIN_B0);
              delay_us(750);
              output_low(PIN_B0);
              delay_ms(20);
       }
```

```
else if(input(PIN_D4)==0 && input(PIN_D3)==0 && input(PIN_D2)==0 &&
input(PIN_D1)==1 && input(PIN_D0)==0)
// input=00010 45 degree left (PanCam1)
{
        output_high(PIN_B0);
        delay_us(1125);
        output_low(PIN_B0);
        delay_ms(20);
}

else if(input(PIN_D4)==0 && input(PIN_D3)==0 && input(PIN_D2)==0 &&
input(PIN_D1)==1 && input(PIN_D0)==1)
// input=00011 0 degree (PanCam1)
{
        output_high(PIN_B0);
        delay_us(1500);
        output_low(PIN_B0);
        delay_ms(20);
}

else if(input(PIN_D4)==0 && input(PIN_D3)==0 && input(PIN_D2)==1 &&
input(PIN_D1)==0 && input(PIN_D0)==0)
// input=00100 45 degree right (PanCam1)
{
        output_high(PIN_B0);
        delay_us(1875);
        output_low(PIN_B0);
        delay_ms(20);
}

else if(input(PIN_D4)==0 && input(PIN_D3)==0 && input(PIN_D2)==1 &&
input(PIN_D1)==0 && input(PIN_D0)==1)
// input=00101 90 degree right (PanCam1)
```

```
{
        output_high(PIN_B0);
        delay_us(2250);
        output_low(PIN_B0);
        delay_ms(20);
}

else if(input(PIN_D4)==0 && input(PIN_D3)==0 && input(PIN_D2)==1 &&
input(PIN_D1)==1 && input(PIN_D0)==0)
// input=00110 45 degree up (TiltCam1)
{
        output_high(PIN_B1);
        delay_us(1875);
        output_low(PIN_B1);
        delay_ms(20);
}

else if(input(PIN_D4)==0 && input(PIN_D3)==0 && input(PIN_D2)==1 &&
input(PIN_D1)==1 && input(PIN_D0)==1)
// input=00111 0 degree (TiltCam1)
{
        output_high(PIN_B1);
        delay_us(1500);
        output_low(PIN_B1);
        delay_ms(20);
}

else if(input(PIN_D4)==0 && input(PIN_D3)==1 && input(PIN_D2)==0 &&
input(PIN_D1)==0 && input(PIN_D0)==0)
// input=01000 45 degree down (TiltCam1)
{
        output_high(PIN_B1);
        delay_us(1125);
        output_low(PIN_B1);
```

```
        delay_ms(20);
}

else if(input(PIN_D4)==0 && input(PIN_D3)==1 && input(PIN_D2)==0 &&
input(PIN_D1)==0 && input(PIN_D0)==1)
// input=01001 90 degree left (PanCam2)
{
        output_high(PIN_B2);
        delay_us(750);
        output_low(PIN_B2);
        delay_ms(20);
}

else if(input(PIN_D4)==0 && input(PIN_D3)==1 && input(PIN_D2)==0 &&
input(PIN_D1)==1 && input(PIN_D0)==0)
// input=01010 45 degree left (PanCam2)
{
        output_high(PIN_B2);
        delay_us(1125);
        output_low(PIN_B2);
        delay_ms(20);
}

else if(input(PIN_D4)==0 && input(PIN_D3)==1 && input(PIN_D2)==0 &&
input(PIN_D1)==1 && input(PIN_D0)==1)
// input=01011 0 degree (PanCam2)
{
        output_high(PIN_B2);
        delay_us(1500);
        output_low(PIN_B2);
        delay_ms(20);
}
```

```
else if(input(PIN_D4)==0 && input(PIN_D3)==1 && input(PIN_D2)==1 &&
input(PIN_D1)==0 && input(PIN_D0)==0)
// input=01100 45 degree right (PanCam2)
{
            output_high(PIN_B2);
            delay_us(1875);
            output_low(PIN_B2);
            delay_ms(20);
}

else if(input(PIN_D4)==0 && input(PIN_D3)==1 && input(PIN_D2)==1 &&
input(PIN_D1)==0 && input(PIN_D0)==1)
// input=01101 90 degree right (PanCam2)
{
            output_high(PIN_B2);
            delay_us(2250);
            output_low(PIN_B2);
            delay_ms(20);
}

else if(input(PIN_D4)==0 && input(PIN_D3)==1 && input(PIN_D2)==1 &&
input(PIN_D1)==1 && input(PIN_D0)==0)
// input=01110 45 degree up (TiltCam2)
{
            output_high(PIN_B3);
            delay_us(1875);
            output_low(PIN_B3);
            delay_ms(20);
}

else if(input(PIN_D4)==0 && input(PIN_D3)==1 && input(PIN_D2)==1 &&
input(PIN_D1)==1 && input(PIN_D0)==1)
// input=01111 0 degree (TiltCam2)
{
```

```
            output_high(PIN_B3);
            delay_us(1500);
            output_low(PIN_B3);
            delay_ms(20);
    }

    else if(input(PIN_D4)==0 && input(PIN_D3)==1 && input(PIN_D2)==1 &&
    input(PIN_D1)==1 && input(PIN_D0)==0)
    // input=10000 45 degree down (TiltCam2)
    {
            output_high(PIN_B3);
            delay_us(1125);
            output_low(PIN_B3);
            delay_ms(20);
    }

    else {}

    }

}
```