

Design and Construction of PIC-based IR Remote Control Moving Robot

Sanda Win, Tin Shein, Khin Maung Latt

Abstract—This document describes an electronic speed control designed to drive two DC motors from a 6 V battery pack to be controlled by a commercial universal infrared remote control hand set. Conceived for a tank-like vehicle, one motor drives the left side wheels or tracks and the other motor drives the right side. As it is shown here, there is a left-right steering input and a forward-backward throttles input, like would be used on a model car. It is designed using a microcontroller PIC16F873A.

Keywords—Assembly Language, Direction Control, Speed Control, PIC 16F 873A

I. INTRODUCTION

INDUSTRY automation is mainly developed around motion control systems in which controlled electric motors play a crucial role as heart of the system. Therefore, the high performance motor control systems contribute, to a great extent, to the desirable performance of automated manufacturing sector by enhancing the production rate and the quality of products. In fact the performance of modern automated systems, defined in terms of swiftness, accuracy, smoothness and efficiency, mainly depends to the motor control strategies. The advancement of control theories, power electronics and microelectronics in connection with new motor designs and materials have contributed largely to the filed of electric motor control for high performance systems.

In this project a small robot which can move with various speeds and directions is designed and constructed. Its movements can be controlled by a universal remote control from a distance of fifteen feet. The robot able to move forward and reverse directions with different speeds and it can rotate to the left and right direction of up to twenty degree. The robot is constructed by using precision DC motor and the control instructions are written by assembly language and preinstalled in a microcontroller. The motor is switched by an H-Bridge of transistors, controlled by the PIC16F873A.

A commercially available remote control is first analyzed and its keys are assigned to each instruction that control the movements of the constructed robots. A total of ten keys are assigned to control the robot's speed and direction.

According to the test results the robot received the control-signal without interference and work precisely as instructed. Construction is very simple, required a few tools and does not involve high-precision engineering.

Department of Physics, Yangon Technological University, Myanmar,
(Corresponding author, Phone: 09-51-81692; e-mail: sandarw75@gmail.com).
Department of Physics, Yangon Technological University, Myanmar.
Department of Physics, Mandalay Technological University, Myanmar.

This robot can be applied in hazardous and dangerous areas to work for human for simple and repeated tasks which required some degree of accuracy and precision.

II. THEORY OF OPERATION

This project was divided into four sections which included:

- 1) *IR Interface* : This block converts the pulse width modulated infrared signal from a standard commercial IR remote control signal to binary data stream.
- 2) *IR Decoder* : This block converts the steering (direction) and throttle (speed) signals into digital signals for direction control logic circuitries and PWM generators.
- 3) *PWM Generator* : Converts control signals from the previous stage into Pulse Width Modulated digital signals suitable for driving the power FET drivers to drive the motors.
- 4) *FET Driver Module and Direction Control Logic Circuitries for Left and Right Motors* : Contain the power FET's and associated bipolar transistors to convert digital control signals into motor drive power.

A. IR Interface

The typical infrared signal used by remote controls has three layers. The names used for these layers have not been standardized. In this paper they are called the infrared, the modulation, and the serial data. The infrared layer is the means of transmission. Infrared is light whose wavelength is too long to see. The modulation layer refers to the fact that each burst of infrared signal is often modulated at a frequency between 32.75 kilo Hertz (kHz) and 56.8 kHz. This is done to diminish the effects of ambient light. The serial data layer has the information containing a command. This is typically coded in the lengths of infrared bursts or in the lengths of gaps between infrared bursts. A long gap or burst is interpreted as a "1", a short gap or burst is interpreted as a "0" [1].

A Sony remote control transmitter is used in this research. The Sony remote control is based on the Pulse-Width signal coding scheme. The code exists of 12 bits sent on a 40 kHz carrier wave. The code starts with a header of 2.4 milli second (ms) or 4 times T, where T is 600 micro second (μ s). The header is followed by 7 command bits and 5 address bits as shown in Fig. 1. The address and commands exists of logical ones and zeros. A logical one is formed by a space of 600 μ s or 1 T and a pulse of 1200 μ s or 2T. A logical zero is formed by a space of 600 μ s and pulse of 600 μ s. The space between 2 transmitted codes when a button is being pressed is 40 ms. The bits are transmitted least significant bits first. The total length of a bit-stream is always 45 ms [2].

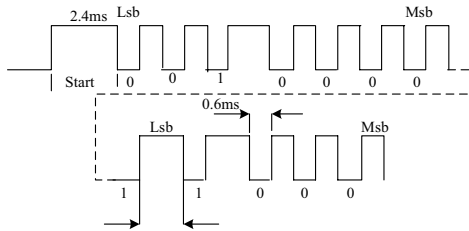


Fig.1 Sony Infrared Remote Protocol

An IR remote control (the transmitter) sends out pulses of infrared light that represent specific binary codes. These binary codes correspond to commands, such as Power On/Off and speed up. The IR receiver in the robot decodes the pulses of light into the binary data (ones and zeroes) that the device can understand. The microcontroller then carries out the corresponding command. The control codes are sent in serial format modulated to that 40 kHz carrier frequency (usually by turning the carrier on and off). There are many different coding systems in use, and generally different manufacturers use different codes and different data rates for transmission. The data rate send is generally infra range of 100-2000 bps.

The only mandatory hardware for decoding IR signals is an infrared receiver. The receiver circuit consists of a photodiode, preamplifier, and a demodulator circuit as shown in Fig. 2. This combination is commercially available as the Sharp GP51UX, and IS1U60x.

The preamplifier contains a band pass filter which limits the receiver's sensitivity to about ±2 kHz, near the centre frequency. An Automatic Gain Control (AGC) circuit adjusts the incoming level to the demodulator, which explains the presence of a long leading pulse in many of the protocols. This allows the receiver to stabilize its AGC circuit, prior to the reception of the bit-stream.

The output of the receiver is a binary bit-stream, corresponding to the original modulation signal at the transmitter. It is often an open collector pull-down. Note that this signal is active low, so that "ones" in terms of the carrier signal appear as "zeros" at the demodulator. A typical block diagram of an IR receiver is shown in Fig. 2.

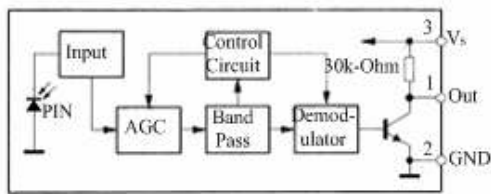


Fig. 2 Block Diagram of the Receiver

This project used PIC16F873A Microcontroller so that some operation can be done in a single chip because IR decoder and PWM generator are residing in the microcontroller. Fig. 3 illustrates the functional block diagram for PIC16F873A microcontroller unit (MCU).

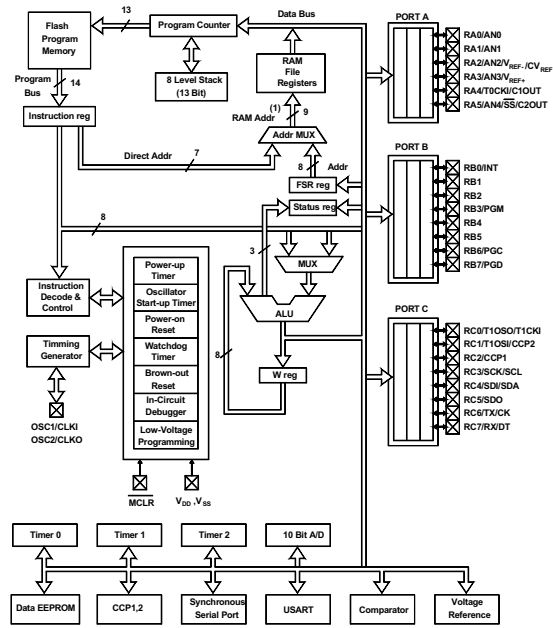


Fig. 3 Internal Block Diagram of PIC16F873A

This is a rather complex IC because it contains an 8 bit central processing unit (CPU) core along with additional features such as:

Microcontroller Core Features:

- 1) 4K words of ROM (Flash Program Memory)
- 2) 128 Bytes of EEPROM
- 3) 192 Bytes of data RAM
- 4) 200 nano second (ns) instruction execution
- 5) Only 35 single word instruction to learn
- 6) 3 Input/Output ports
- 7) High sink/source current: 25 milli Ampere (mA).

Peripheral Features:

- 1) Timer0
- 2) Timer1
- 3) Timer2
- 4) Two Capture, Compare, PWM modules (PWM maximum resolution is 10 bit)
- 5) 10 bit 5 channel Analogue to Digital converter
- 6) Synchronous Serial Port (SSP)
- 7) Universal Synchronous Asynchronous Receiver Transmitter (USART)
- 8) Parallel Slave Port [3].

B. IR Decoder

IR Decoder is a software module and it is written in the Flash Program memory of the PIC16F873A microcontroller. The main task of the decoder is that if a button is pressed on a remote control hand set, the software program translates the codes received to the actual button pressed. Serial Infrared Control (SIRC) protocol is the name given to Sony's IR remote control. The 12 bit protocol is the most common format used with domestic products.

As previously mentioned, the common word is made up of 12 bit, and consists of a 7 bit command code followed by a 5 bit device code, see Fig. 4. This SIRC format uses pulse width modulation of the infrared signal to transmit the data. The SIRC transmission is preceded by a single start bit. The decoding software waits for this start bit of 2.4 ms. Using a unique signal as a start bit helps prevent the software trying to decode an incomplete transmission. The infra-red sensor uses this start pulse to set its AGC. When it is correctly received a software flag is set 1 to allow the rest of the transmission to be decoded. The SIRC data consists of either 0.6 ms or 1.2 ms duration, meaning logic 0 and 1 respectively. Each pulse is preceded by a 0.6 ms pause.

The pulse length is measured by polling the falling edge of the waveform using the build-in hardware timer, Timer 0. With 4 MHz crystal oscillator and prescalar values of 16, the timer value is incremented every 16 μs and is read on every falling edge of the waveform.

In order to work out the likely timer values, by using the following formula, divide the expected pulse width by the timer:

$$\frac{\text{pulse width}}{\text{timer}} = \frac{\text{start pulse}}{16\mu\text{s}} = \frac{2.4\text{ms}}{16\mu\text{s}} = 150 \quad (1)$$

The program uses the timer value to determine the waveform. For example, if the value is between 90 and 150 then a logic 1 is assumed and if the value is between 50 and 90 then a logic 0 is assumed (see table I).

| | |
|---------------|-----------------|
| 2.4ms | = 150 (start) |
| 2.4ms + 0.6ms | = 187 (start) |
| 1.2ms + 0.6ms | = 112 (logic 1) |
| 0.6ms + 0.6ms | = 75 (logic 0) |

Note that the remote control may generate different numbers for the same function so that the Sony equipment can distinguish between, for example, Play for the CD player and Play for the tape recorder.

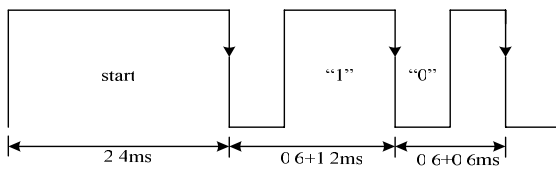


Fig. 4 SIRC Timing Details

Certain remote control key codes are recognized by the IR decoder software and used to control the speed and direction of rotation of the left and right motor. The following table shows the remote control keys codes and corresponding function implement by the IR decoder software [4].

TABLE II
REMOTE CONTROL FUNCTIONS ASSIGNED TO CONTROL ROBOT MOVEMENT

| Key | Sony function | Robot function |
|-----|-------------------------|----------------------------------|
| 1 | 1 (channel selector) | start forward with minimum speed |
| 2 | 2 | stop |
| 3 | 3 | speed decrease |
| 4 | 4 | speed increase |
| 5 | 5 | turn right |
| 6 | 6 | turn left |

C. Pulse Width Modulation (PWM) Generator

Pulse Width Modulation is critical to modern digital motor controls. By adjusting the pulse width, the speed of a motor can be efficiently controlled without larger linear power stages. Some PIC devices and all dsPIC DSCs have hardware PWM modules on them. These modules are built into the Capture/Compare/PWM (CCP) peripheral. As previously mentioned PIC16F873A has two CCP modules. Each CCP module is software programmable to operate in one of three modes:

- 1) A Capture input
- 2) A Compare output
- 3) A Pulse Width Modulation (PWM) output

For the CCP module to function, Timer resources must be used in conjunction with the CCP module. The desired CCP mode of operation determines which timer resources are required.

TABLE III
CCP MODE-TIMER RESOURCE

| CCP MODE | Timer Resource |
|----------|----------------|
| Capture | Timer 1 |
| Compare | Timer 1 |
| PWM | Timer 2 |

PWM Mode:

A Pulse Width Modulation output (Fig. 5) is a signal that has a time-base (period) and a time that the output stays high (duty cycle). The period is the duration after which the PWM rising edge repeats itself. The resolution of the PWM output is the granularity with which the duty cycle can be varied. The frequency of a PWM is simply the inverse of the period (1/period).

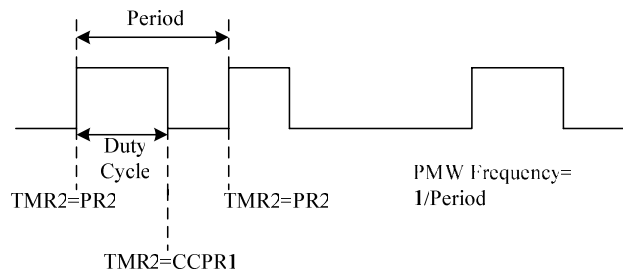


Fig. 5 PWM Output

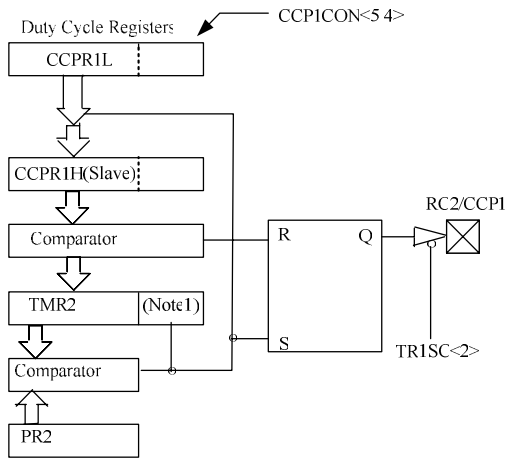


Fig. 6 PWM Mode Block Diagram

Each CCP module can support one Pulse Width Modulation (PWM) output signal, with minimal software overhead. This PWM signal can attain a resolution of up to 10 bits, from the 8 bit Timer 2 module. This gives 1024 steps of variance from an 8 bit overflow counter. This gives a maximum accuracy of T_{osc} (50 ns, when the device is operated at 20 MHz). Fig. 6 shows a block diagram of the CCP module in PWM mode. When the Timer 2 overflows (timer=period register), the value in the duty cycle registers (CCPRxL:CCPRx-CON<5:4>) is latched into the 10 bit slave latch. A new duty cycle value can be loaded into the duty cycle register(s) at any time, but is only latched into the slave latch when $Timer\ 2 = Timer\ 2\ Period\ Register\ (PR\ 2)$. The period of Timer 2 (and PWM) is determined by the frequency of the device, the Timer 2 prescaler value (1, 4 or 16), and the Timer 2 Period Register.

The following two equations show the calculation of the PWM period, and duty cycle [5].

$$PWM\ Period = [(PR\ 2) + 1] \cdot 4 \cdot T_{osc} \quad (2)$$

(Timer 2 prescaler value)

$$PWM\ Duty\ Cycle = [CCPRxL:CCPRxCON<5:4>] \cdot 4 \cdot T_{osc} \quad (3)$$

(Timer 2 prescaler value)

D. FET Driver Module and Direction Control Logic Circuitries for Left and Right Motors

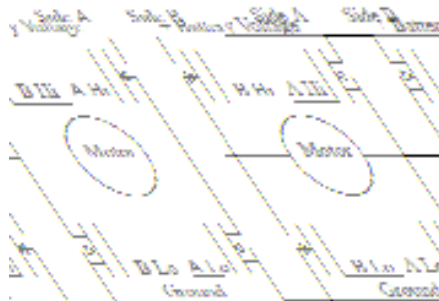


Fig. 7 H-Bridge Motor Drive Circuit

Since the motors must be able to run forward or backward, it is command method to set the output transistors up in an H-Bridge configuration as shown in the Fig. 7. This is the only solid state way to operate a motor in both directions. In this configuration, each side of the motor gets two transistors attached to it: one tied to the battery positive line and the other tied to ground. It is obvious that both transistors on side A can not be on at the same time and the same applies to side B, so Side A Lo can be driven off of an inverted copy of Side A Hi and Side B Hi can be driven off of an inverted copy of Side B Lo. This arrangement means that the high and low transistors can never be on at the same time and it is required to generate two unique control signals per motor now. In order to run the motor forward, it is required to turn on transistors on the Side A Hi and the Side B Lo. For reverse, it is required to turn on the Side B Hi and the Side A Lo transistors [6].

The complete circuit is shown in Fig.8. This circuit is powered by a 6 V battery via switch S2 and diode D2. D2 serves a dual purpose-first, to prevent reverse polarity, which could do considerable damage, and second, to drop the supply voltage to about 5.4 V, which is more suitable for the PIC16F873A. There are two identical H-Bridge motor drives, one for the left motor and one for the right.

Pin 25 (RB4) and pin 23 (RB2) of the microcontroller is designated by the manufacturers for input or output. In this circuit, they are used for output only. Pin 25 is used for control the direction (forward or reverse) of the left hand motor, as seen from the rear of the robot. Pin 23 is used here to control the direction (forward or reverse) of the right hand motor.

Direction control logic signals are from RB4 and RB2 of microcontroller switch two power MOSFETs H-Bridges to control the direction of the motors (forward or reverse). The two 100 (nF) capacitors across each motor.

Pin 13 (RC2) of PIC16F873A can be used as a general purpose I/O pin or I/O pin for CCP1 module. In this circuit it is configured as CCP1 pin to produce PWM output. It is used to switch both of the motors on or off at the same time. It is also used to produce PWM for speed control of both motors. When it is 'high', the motors are on: when it is 'low' they are off.

Since two logically inverted control signals are required for each side of an H-Bridge, a BJT transistor has been added in each H-Bridge motor drives circuit (Q1 for motor 1 and Q8 for motor 2). Actually transistors Q1 and Q8 are used as inverters so that when the "forward motion" MOSFETs Q3 and Q4 are disabled, the "reverse motion" MOSFETs Q6 and Q7 are activated.

Note that neither pin 25 nor pin 23 will accomplish anything unless both motors are switch on first via pin 13 (RC2). Both pins 25 and 23 cause a wheel to rolls forward when it is high and backwards when it is "low". Pins 13, 25, and 23 together may be used not only to make the robot drive forwards or reverse but also to turn right or left.

Pin 13 (RC 2) activates both motors simultaneously via MOSFETs Q2 and Q5. These two MOSFETs are wired in parallel and these should work satisfactorily with a small heat sink for the small motors used here.

Referring back to the drawing of the H-Bridge, it can be seen that if both transistors on one side of a bridge were turned on at the same time, it would have a direct short to ground.

This problem is called shoot through current and it is a bigger problem than might be expected. FET's have a lot of capacitance 2000 pico Farad (pF) for the one used in this circuit on their gate leads, so it is difficult to switch them on or off quickly. This switching delay makes it very easy to have both FET's on for a short period of time each time. There is a transition from one FET conducting to the other. A lot of power can go through in that time and it will heat up transistors and cook them very quickly if allowed to happen.

For this reason both motors are switched off first via pin 13, whenever one of the motor is need to change direction of rotation. After the required direction control commands are sent to the H-Bridges via pin RB4 and RB2.

The two motors are switched on again via pin 13 and the previous PWM output is routed to the FET driver transistors Q2 and Q5.

III. CONCLUSION

After designing and testing this project, it became pretty obvious that microcontroller is a smart way to do this due to the high part count and the somewhat involved set up and calibration procedure, but there is still value in having a controller that can be built with no exotic equipment. An additional feature of this design is that it can be built in several configurations due to the modularity of the design.

REFERENCES

- [1] William G. Grimm, "Decoding Infrared Remote Controls Using a PIC 16C5x Microcontroller", 1997.
- [2] David De Vleeschauwer, "DAVshomepage" <http://user.pandora.be/davshomepage>
- [3] Microchip. "PIC16F873A Data Sheet" Microchip Technology Inc. May, 2001.
- [4] Roger Thomas, "Remote Control IR Decoder" Everyday Practical Electronics, September, 2000. UK. <http://www.microchip.com>
- [5] Mark Palmer, "Using the CCP Modules" Microchip Technology Inc. 1997. <http://www.microchip.com>
- [6] Bob Harbour, "Dual Motor Bidirectional Electronic Speed Control", April 30, 1999. <http://www.circuits.Lab.com>

