UNITECH '08 GABROVO
INTERNATIONAL SCIENTIFIC CONFERENCE
21 – 22 November 2008, GABROVO

POSITION CONTROLS OF 4 SERVO MOTORS WITH POTENTIOMETER, BY USING MICROCONTROLLER

ÇETİNKAYA. Ö[1]
KUŞÇU. H[2], TOYLAN.H[3]

[1]Trakya University, Science Institute, Mechanical Engineering Division
MBS Student, Turkey, ozcan@cetinkaya.gen.tr

[2]Trakya University, Engineering and Architecture Faculty, Mechanical Engineering Division,
Mechanical Theory and Dynamics Main Branch, Turkey, hilmi@trakya.edu.tr

[3]Kırklareli University, Turkey, hayrettintoylan@trakya.edu.tr

Abstract

In this study, position controls of 4 Hobby Servo motors with potentiometer have been done by using microcontroller. Position control of each motor has been done according to the analogue duty that reaches from potentiometers which move independently from each other. The axis of Hobby Servo motor cycles between 0-180 degrees. This cycling gets an angle according to modulation width between 1-2ms of 50Hz frequency signal that is applied to control point. Analogue signal which comes from potentiometers are transformed to numerical data (value) as modulation width in Analogue-Digital Transformer of microcontroller, this numerical data is manipulated in microcontroller again and is retransformed to a 50 Hz frequency signal that modulation width changes between 1-2ms. This study can be used to budge the arms of robot which has four or more independency degree.

Key Words: Servo Motor, Position Control, Microcontroller

INTRODUCTION

Nowadays, by decreasing the cost of microcontroller and making the microcontroller softwares by high level programming languages have increased the use of microcontrollers in many areas. In this study, axis position of servo motor which are used as hobby is going to be controlled by using microcontroller with potentiometer. For this purpose, PIC16F877/A microcontroller which has analogue-digital converter (ADC) is used to process the analogue signal that comes from potentiometer. Analogue signal that comes from potentiometer is converted to an 8 bit digital data. This digital data is converted to a digital output that changes linearly between 1-2 ms in modulation width, 50 Hz frequency from data outputs by software that loaded to microcontroller. So the axis of servo motor gets an angle value between 0-180° according to analogue signal which comes from potentiometer.

EXPOSITION

Hobby servo motors are used to act the sensitive parts as arm, wing, and propeller of plane, helicopter, ship, robot and model vehicles like these. (Figure-1)

Figure-1. A typical hobby servo motor.

Because it is used rather frequently for model vehicles, known as RC (radio control) servo motor in market, too. They get a high torque with a low voltage between 4-6V by courtesy of reduction gear interior part of servo motors. The axis of servo motors does not cycle continually as the axis of right movement or alternative movement motor. There are three control pin where control signal is applied to determine axis position. The axis of servo motor gets an angle between 0-180° according to the signal that is applied to control pin. If the right movement is only applied to feeder pins of servo motor, it doesn't move. To get angle for servo motor axis, it is needed to apply a signal to control point which changes 1-2ms modulation width and 50Hz frequency.

Angle of axis changes between 0-180° proportionally with 1-2ms modulation width. For example, the axis gets 0° angle value with an 1ms modulation width signal (figure-2a), 1,5ms modulation with 90° (figure-2b) and 2ms modulation width 180° (figure-2c)

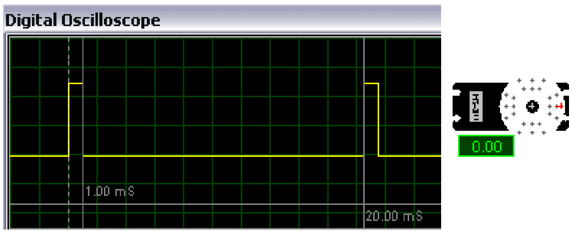

Figure 2a

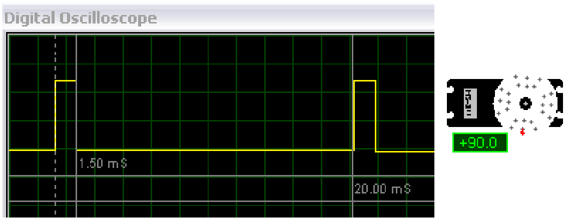

Figure 2b

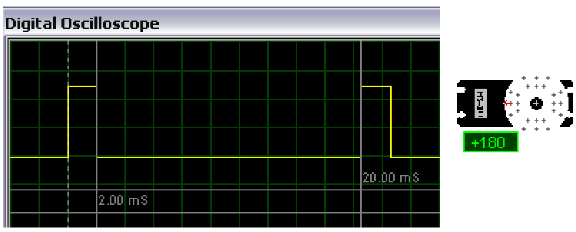

Figure 2c

Figure 2a, 2b, 2c Angle values of motor axis according to different modulation width.

In this study potentiometer has been used to change the angle of motor axis. It is seen in figure-3 that collective pin has been used as voltage divider by applying +5v voltage at both stable pins of potentiometer. According to the angle of setup pin of potentiometer, an analogue signal between 0 and +5v is got proportionally from middle pin.

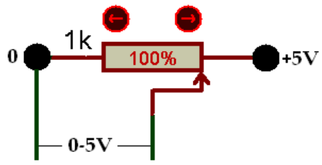

Figure-3.Usage of potentiometer as voltage divider.

16F877/A microcontroller has been used to convert the analogue signal which is received from potentiometer at a signal which has modulation width to control servo motor.

16F877/A microcontroller has 8 analogue-digital converter which has 10bits resolution and 25ends which can be used as digital input or output that is seen in figure-4.

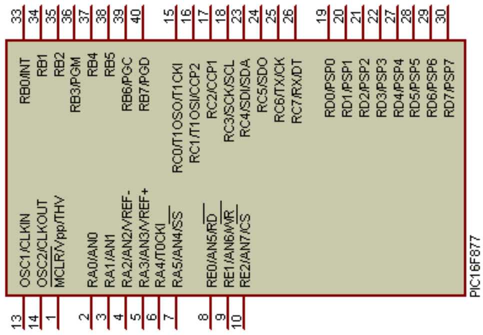

Figure-4, 16f877/a microcontroller's foot names.

The pins of analogue-digital converter are in order RA0/AN0, RA1/AN1, RA2/AN2, RA3/AN3, RE0/AN4, RE1/AN5, RE2/AN6 and RE3/AN7 pins. In this study, PORTA's primitive four pins have been used to convert the analogue signal to digital signal which is received from potentiometer.

Due to drive servo motors RB0, RB1, RB2 and RB3 pins of PORTB have been attended and used as output pin. Conception of electronic circuit used in this study has been prepared by Proteus ISIS programme as in figure-5.

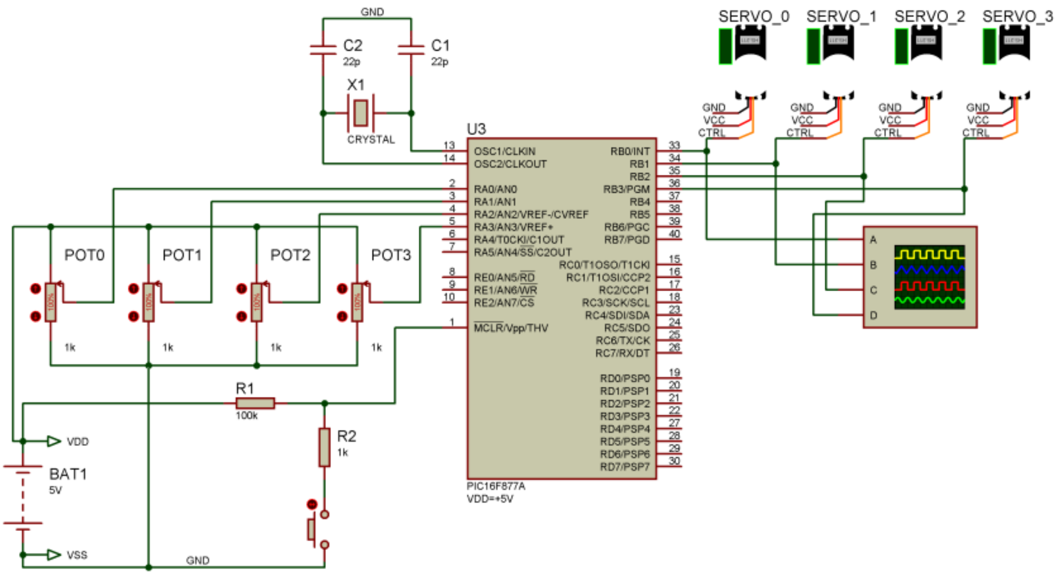

Figure-5, electronic circuit which is prepared by ISIS.

SOFTWARE

It is needed to load a software to microcontroller for preparing simulation of electronic circuit which is prepared by ISIS and for running it in real life. That software must be enough for needs in our study so that our circuit can work correctly and orderly. We need to determine the programming language that we will use before starting to prepare the software. Therefore assembler PICBASIC, CCS C and some programmers like this can be used. We have prepared software with PICBASIC which is a mid-level programming language in our study. First of all we must determine that analogue data is going to convert to digital data which comes from potentiometer with how much bits resolution; we must determine clock frequency of oscillator which is used in electronic circuit. Therefore following definitions are given in order.

DEFINE ADC_BITS 8; bit of resolution is determined

DEFINE ADC_CLOCK 3; clock source is determined. (Internal Oscillator)

DEFINE ADC_SAMPLEUS 100; exemplifying is made in 100microseconds periods

DEFINE OSC 10; frequency of oscillator is determined. (10 MHz)

Later, variables are determined. In this study, there are 4 inputs and 4 outputs variables an done periodic determine in total 9 variables. ADC_BILGI_? Are input variables and 8 bit variables of analogue data which comes from potentiometer? SERVO_? Variables are output ones and they are the variables of modulation width which determines the axis position of each servo motor. PER variables define the period of formed signal. Variables are defined as WORD type. (-32768, +65536)

ADC_BILGI_0 VAR WORD
ADC_BILGI_1 VAR WORD
ADC_BILGI_2 VAR WORD
ADC_BILGI_3 VAR WORD
SERVO_0 VAR WORD
SERVO_1 VAR WORD
SERVO_2 VAR WORD
SERVO_3 VAR WORD
PER VAR WORD

It is needed to determine the pins which will be used for input and output after variables are defined. If one of the part's related bit of TRIS register is loaded 1, related pin of this port is input pin, if 0 is loaded, it is define as output pin. So 1 must be loaded pre-4 bits of PORTA's TRISA register, 0 must be loaded pre-4 bits of PORTB's TRISB register.

TRISA=%00001111 ; 4 primitive pins of PORTA is defined as input pin.

TRISB=%00000000 ; all pins of PORTB is defined as output pin. (It shows numerical system for % binary)

ADC's ADCON1 register must be set after all definitions have been done.

ADCON1=%00000000; ADCON1 register is set according to our study.

Here pre-4 bit is used to determine that; which of the pins are going to be used for analogue input. We have defined it as 0000 that means these 8 analogue inputs are going to be used. Flowing two bit is not used and must be 0, the sixth bit is used to choose clock frequency, here we used internal oscillator frequency, the last, 7th bit determines the data that is read from analogue inputs to be based on right and left sides. We choose the right based one because we have used 8 bit resolution. Now, it is time to make analogue-digital change an to write the programme which is going to be changed modulation width. Analogue data which comes from potentiometer is transformed to an 8 bit value between 0-255 in microcontroller. This data is read from input pins by the order of ADCIN. In the software BAK: IF ADCON0.2=1 THEN BAK orders are read in an order and then procedure goes on with the flowing orders. Order of ADCIN is used as "analogue input part ", ADCIN.

For example; order of ADCIN 0, ADC_DATA is 0. Analogue data which is read from analogue input pin is attended to ADC_DATA_0 variable. It is needed to transfer this data to output as modulation width. Therefore PULSOUT order is used. PULSOUT order produces Logic-1 signal from determined pin within determined time. "PULSOUT pin is used as waiting time." Here, waiting time (process) is a kind of ms and changes according to the oscillator frequency used in microcontroller. Working process of PULSOUT order in 4 MHz oscillator frequency is 10µs. Now let's make a short calculation. Oscillator frequency is always quadruple of order frequency. Oscillator frequency in our circuit is 10 MHz so working process of microcontroller is $(10*4)/10 = 4\mu s$. It is seen here that each value that we will write on waiting process at PULSOUT order is going to cause us 4µs lateness. Lateness between 1-2ms will be enough for us to drive servo motor. So, let's make a formula which will give us 1ms modulation width when the data is 0 which comes from ADC and 2ms modulation width when the data is 255.
Our formula is going to be as following.

SERVO_?=((ADC_BILGI_?/4)*4)+250 (1)

If we think 255 data comes from ADC, (ADC_BILGI_?=255)

$$\left(\left(\frac{255}{4}\right)*4+250*4\right)*4 = 2020\mu s \quad \cong 2000\mu s \quad (2)$$

Is reached, and it is attended to SERVO_? Variable.

If we think 0 data comes from ADC, (ADC_BILGI_? =0)

$$\left(\left(\frac{0}{4}\right)*4+250\right)*4 = 1000\mu s \qquad (3)$$

Is reached, and it is attended to SERVO_? Variable. (Command cycle 4µs)

In the second formula, although the result is 2020, 20µs is not important because of lateness.

According to the 1st formula, a modulation width between 1-2ms is got directly for any value between 0-255 which comes from ADC. Oscillator that we have used in this study is 10Mhz. Formula 1 must be put in an order whether a different frequency oscillator is going to be used.

In the table below, formulas which we are going to use for different oscillator frequencies are given.

Oscillator Frequency	**A command cycle**	**Formula which is going to be used**
1 MHz	$(10*4)/1 = 40\mu s$	((ADC_BILGI_?/40)*4)+25
4 MHz	$(10*4)/4 = 10\mu s$	((ADC_BILGI_?/10)*4)+100
10 MHz	$(10*4)/10 = 4\mu s$	((ADC_BILGI_?/4)*4)+250
20 MHz	$(10*4)/20 = 2\mu s$	((ADC_BILGI_?/2)*4)+500

Table-1, formula which is going to be used for different SERVO_? Variables in various oscillator frequencies.

The result which is got from the first formula that is used for modulation width is attended to SERVO_? Variable.

The order below provides to produce output from related pins of PORTB width PULSOUT order of SERVO_? Variable.

PULSOUT PORTB.?,SERVO_? ; here logic-1 output is produced from PORTB's ?pin as multiple of waiting process(4µs in this study) of PULSOUT order in SERVO_? Variable value.

For example, if 400 comes to SERVO_0 variable from ADC in PULSOUT PORTB.0, SERVO_0 order line, (400*4µs) 1600 µs Logic–1 output is produced from PORTB's 0 pin and after the process again PORTB's 0, pin becomes Logic-0. Due to protect the position of motor axis or take an other angle, modulation width that is taken from 0, 1, 2 and 3rd pins of PORTB must continue periodically. The frequency of the signal that is sent to control input of servo motors must be 50Hz. So period of the signal that is sent must be $T = 1/50 = 0,02s = 20ms$. After logic-1 output is produced from related pins of PORTB by PULSOUT order in microcontroller software, it is waited by PAUSE order and provided to repeating output signals periodically; usage of pause order is like following.

PAUSE waiting process; here waiting process value is multiplied with 1ms and waits as result process, be continued after the process.

SERVO_0, SERVO_1, SERVO_2 and SERVO_3 modulations take 2020 in total. When they take the maximum value, and this is $2020 * 4\mu s \cong 8ms$.

If we multiply the sem of immediate values of SERVO_0, SERVO_1, SERVO_2 and SERVO_3 variables to 4µs and if we divide them to 1000, we get total modulation width with ms type.

Due to make a 20ms period, per variable is loaded to waiting process of PAUSE order. PER variable is found by abstracting the total modulation width from period process (20ms). Following order line is used in our software to get 20ms period.

PER=(20-(((SERVO_0+SERVO_1+SERVO_2+SERVO_3)*4)/1000))

PER=(20-((SERVO_0+SERVO_1+SERVO_2+SERVO_3)/250)) (4)

PAUSE PER ; so then, period of signals which are sent to control inputs of servo motors are being fixed at 20ms.

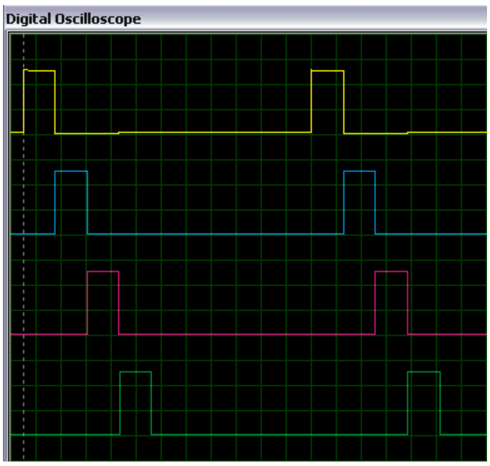

Figure-6, Oscilloscopic view of servo motors' control signals in ISIS simulation programme.

The whole programme which is prepared by PicBasicPro language that is going to be loaded at 16F7877/A microcontroller, is below.

```
TRISA=%00001111
TRISB=0
TRISC=0
TRISD=0
'-------------------------------------------
@ DEVICE pic16F877
@ DEVICE pic16F877, WDT_ON
@ DEVICE pic16F877, PWRT_ON
@ DEVICE pic16F877, PROTECT_OFF
@ DEVICE pic16F877, HS_OSC
'-------------------------------------------
DEFINE    ADC_BITS     8
DEFINE    ADC_CLOCK    3
DEFINE    ADC_SAMPLEUS    100
DEFINE    OSC          10
'-------------------------------------------
ADCON1=%00000000
'-------------------------------------------
ADC_BILGI_0  VAR    WORD
ADC_BILGI_1  VAR    WORD
ADC_BILGI_2  VAR    WORD
ADC_BILGI_3  VAR    WORD
SERVO_0    VAR    WORD
SERVO_1    VAR    WORD
SERVO_2    VAR    WORD
SERVO_3    VAR    WORD
PER        VAR    WORD

BASLA:
ADCIN 0, ADC_BILGI_0
ADCIN 1, ADC_BILGI_1
ADCIN 2, ADC_BILGI_2
ADCIN 3, ADC_BILGI_3

BAK:   IF ADCON0.2=1 THEN BAK

SERVO_0=((ADC_BILGI_0/4)*4)+250
SERVO_1=((ADC_BILGI_1/4)*4)+250
SERVO_2=((ADC_BILGI_2/4)*4)+250
SERVO_3=((ADC_BILGI_3/4)*4)+250
PER=(20-
((SERVO_0+SERVO_1+SERVO_2+SERVO_3)/250))

PULSOUT PORTB.0, SERVO_0
PULSOUT PORTB.1, SERVO_1
PULSOUT PORTB.2, SERVO_2
PULSOUT PORTB.3, SERVO_3
LOW PORTB.0
LOW PORTB.1
LOW PORTB.2
LOW PORTB.3

PAUSE PER

GOTO BASLA
```

CONCLUSION

In our study, position control of hobby servo motor axle by using microcontroller and potentiometer has been explained. This study is also carries a pre-study quality for a robot arm which has a turning system. A basic robot arm which has 4 free degree can be made with this study. Microcontroller that is used has 8 analogue-digital converters so its free degree can be increased.

REFERENCE

[1] ALTINBAŞAK, O., 2007. PicBasic Pro ile Pic Programlama. Altaş Yayıncılık, Istanbul

[2]http://ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf

[3] ÇETİNKAYA Ö., KUŞÇU H. "Classification of the Industrial Robot Arms", International Scientific Conference AMTECH'07 Gabrovo, Advanced Manufacturing Technologies Proceedings, Volume-II, ISBN 978-954-683-383-9, pp.I_126-I_131, 23-24 November, 2007, Bulgaria