

```

// =====
//
// Purpose.... Servo Driver, ATMEGA8
// Author..... (C) 2010 - Brian
// E-mail..... admin@codemonkeyrawks.net
//
// =====

// -----[ Program Description ]-----

// Using Serial Commands, Set the Servos to Add, Subtract or Hold Position

// -----[ Variable Arrays ]-----

#include <stdio.h>
#include <avr/io.h>
#include <stdlib.h>
#include <avr/pgmspace.h>
#include <avr/interrupt.h>

#define FOSC 8000000
#define F_CPU 8000000
#define BAUD 9600
#define MYUBRR (((FOSC * 10) / (16L * BAUD)) + 5) / 10) - 1)
#define sbi(var, mask) ((var) |= (uint8_t)(1 << mask))
#define cbi(var, mask) ((var) &= (uint8_t)~(1 << mask))

char Run;

// -----[ Functions ]-----

void ioinit(void);           //Initializes IO
void PosValue(void);         //Check IO
void CScreen(void);          //Clear Screen
uint8_t uart_getchar(void);  //Get Character, Serial
void Servos(void);           //Pulsing IO, Servos
void delay_ms(uint16_t x);   //General purpose delay
static int uart_putchar(char c, FILE *stream); //Put, Serial Data

static FILE mystdout = FDEV_SETUP_STREAM(uart_putchar, NULL, _FDEV_SETUP_WRITE);

uint8_t Data;                // Input
uint8_t Time = 13;           // Servo

// -----[ Initialization ]-----

int main (void)
{
    ioinit();                 //Setup IO pins and defaults

// -----[ Main Program ]-----

Run:

    Data = uart_getchar();

    if(Data == '0')
    {
        if (Time < 15)
        {
            Time = Time + 1 ;
        }
    }

    if(Data == '5')
    {
        if (Time > 11)

```

```

        {
            Time = Time - 1 ;
        }
    }

    Servos();                //Pulsout, Servos
    Servos();                //Pulsout, Servos
    Servos();                //Pulsout, Servos
    Servos();                //Pulsout, Servos
    Servos();                //Pulsout, Servos
    Servos();                //Pulsout, Servos
    Servos();                //Pulsout, Servos
    Servos();                //Pulsout, Servos

    goto Run;
}

// -----[ Subroutines ]-----

void Servos (void)
{
    int i;

    for (i = 0; i < 2; i++)    //Send 50 pulses with width 500 uS

    {
        PORTC= 0x60;          //Pulse high for 2.1mS
        delay_ms(Time);
        PORTC= 0x00 ;         //Pulse low for 20mS
        delay_ms(160);
    }
}

void ioinit (void)
{
    //1 = output, 0 = input
    DDRB = 0b11101111; //PB4 = MISO
    DDRC = 0b00100000; //All Inputs, Except 6
    DDRD = 0b11111110; //PORTD (RX on PD0)

    //USART Baud rate: 9600
    UBRRH = MYUBRR >> 8;
    UBRRL = MYUBRR;
    UCSRB = (1<<RXEN)|(1<<TXEN);
    UCSRC = (1<<URSEL)|(3<<UCSZ0);

    stdout = &mystdout; //Required for printf init
}

static int uart_putchar(char c, FILE *stream)
{
    if (c == '\n') uart_putchar('\r', stream);

    loop_until_bit_is_set(UCSRA, UDRE);
    UDR = c;

    return 0;
}

//Get 8-Bit ASCII II
uint8_t uart_getchar(void)
{
    while( !(UCSRA & (1<<RXC)) );
    return(UDR);
}

//General short delays
void delay_ms(uint16_t x)
{
    uint8_t y, z;

```

```
for ( ; x > 0 ; x--){  
    for ( y = 0 ; y < 90 ; y++){  
        for ( z = 0 ; z < 6 ; z++){  
            asm volatile ("nop");  
        }  
    }  
}
```