```
int i = 0;
int Leitura=0;

//-------------------------------------------------------------------------------//

void Delta_Sigma_Init(){
  VRCON = 0xEC;
  TRISA3_bit = 0;
  CMCON = 0x06;
}
//-------------------------------------------------------------------------------//

int Delta_Sigma_Leitura (int Valor){
 int ADC_Valor = 0;

  T0CS_bit = 0;
  PSA_bit = 1;
  CMCON = 0x03;

  while (Valor){

    TMR0 = 0xF7;
    T0IF_bit = 0;

    if (C1OUT_bit){
      RA3_bit = 1;
    }
    else{
      RA3_bit = 0;
      ADC_Valor++;
    }
    Valor--;
    while (!t0if);
  }

  CMCON = 0x06;
  return (ADC_Valor);
}
//-------------------------------------------------------------------------------//

//Numero maximo de servos
const N_SERVOS = 8;

//Tick = 4 / clock(16MHz) * Prescaler
//Tick = 4 / 16 * 4 = 1us
const unsigned TicksFrame =  2500;   // 20 ms / 8 servos / Tick
const unsigned TicksMin = 1000;      // posicao 0° = 1 ms / Tick
const unsigned TicksCenter =  1500;  // posicao 90° = 1.5 ms / Tick
const unsigned TicksMax = 2000;      // posicao 180° = 2 ms / Tick

unsigned Timer1 at TMR1L;

//Registro de configuracao para cada servo
typedef struct
{
    char Port;
    char Pino;
    unsigned Angulo;
    union
    {
       char Enable:1;
```

```
    };
}Servos;

//cria os 8 servos
Servos myServos[N_SERVOS];

static char flags = 0;
    sbit pulso at flags.B0;

//Inicializa os servos
void Servo_Init()
{
char i;
static char temp = 0;
     for( i = 0; i < N_SERVOS; i++ )
     {
         myServos[i].Port = &temp;
         myServos[i].Pino = 0;
         myServos[i].Angulo = TicksMin;
         myServos[i].Enable = 0;
     }

     Timer1 = 65535 - TicksFrame;
     T1CON = 0b00010001; //Clock 8 Mhz / Prescaler 1:2
     TMR1IE_bit = 1;
     PEIE_Bit = 1;
     GIE_Bit = 1;
}

//Adiciona um novo servomotor
void Servo_Attach( char servo, char out, char pin )
{
Servos *ptr;
      if( servo > N_SERVOS )
         return;
      ptr = &myServos[servo];
      (*ptr).Port = out;
      (*ptr).Pino = 1 << pin;
      (*ptr).Enable = 1;
}

void Servo_Interrupt()
{
static char servo_idx = 0;
char port, pino;
unsigned an;

        //Passa o valor do endereço para o ponteiro
        FSR = (char)&myServos[servo_idx];

        port = INDF; //Recebe o valor da porta, myServos[x].Port
        FSR++;       //Incrementa o ponteiro
        pino = INDF; //Recebe o valor do pino, myServos[x].Pino
        FSR++;       //Incrementa o ponteiro
        ((char*)&an)[0] = INDF;
        FSR++;
        ((char*)&an)[1] = INDF;//Recebe o angulo, myServos[x].Angulo
        FSR++;

        //Servo habilitado?
        if( INDF.B0 ) //myServos[x].enable
        {
```

```
            FSR = port; //Passa para o ponteiro o endereço da porta

            if( !pulso )
            {
              Timer1 = 65535 - an + 29;//(29) - fator de correção?
              INDF |= pino;
            }
            else
            {
              Timer1 = (65535 - TicksFrame) + an + 37;
              INDF &= ~pino;
              ++servo_idx &= 7;
            }

            pulso = ~pulso;
        }
        else
        {
            Timer1 = (65535 - TicksFrame);
            ++servo_idx &= 7;
        }
}



void interrupt()
{
    if( TMR1IF_Bit )
    {
       Servo_Interrupt();
       TMR1IF_Bit = 0;
    }
}

void main()
{
     // TRISA = 0b00100001;
      TRISB = 0; //Define os pinos da PORTB como saída
      //uart1_init(9600);//inicia o modulo UART com velocidade de 9600bps
      Delta_Sigma_Init();
      Servo_Init();
      Delay_ms(200);

      //UART1_Write_Text(" UART Test Successful! ");   // Character Message to be
 Sent



      Servo_Attach( 0, (char)&PORTB, 0 );
      Servo_Attach( 1, (char)&PORTB, 1 );
      Servo_Attach( 2, (char)&PORTB, 2 );
      Servo_Attach( 3, (char)&PORTB, 3 );

      while(1)
      {
      Leitura = Delta_Sigma_Leitura(1900);

      if(porta.f5==1){
      delay_ms(100);
      if(porta.f5==1){
      i++;
      if (i > 3) i = 0;
```

```
        }
        }

        myServos[i].Angulo = (500 + Leitura);
        delay_ms(10);

        }

}
```