UNIVERSIDADE DE BRASÍLIA - UnB

Faculdade do Gama - FGA

Eletrônica Embarcada (120871)

PONTO DE CONTROLE 4

Projeto: Mini cortador de grama controlado por MSP430 (Minigrama)

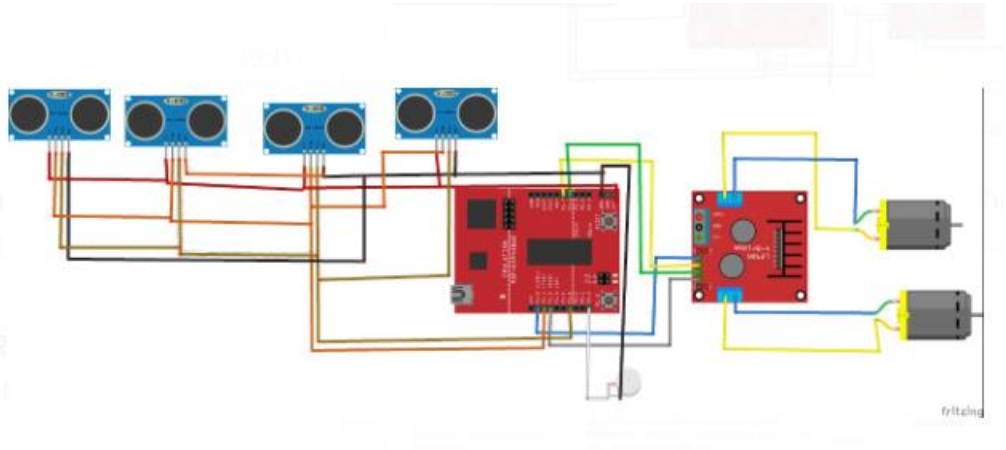
Débora Janini Campos Guedes
Matrícula: 15/0008619
E-mail: deborajaninicg@gmail.com

Maysa Paula Lacerda Cardoso
Matrícula:11/0133315
E-mail: maysa41@hotmail.com

Atividades desenvolvidas e melhoradas entre os pontos de controle 3 e 4, como solicitado em sala (um documento só com as alterações, separado dos demais pontos de controle).

- O superaquecimento da Ponte H, que acabava por queimá-la, foi resolvido adicionando um cooler para resfriamento do componente, fazendo assim o projeto funcionar perfeitamente conseguindo controlar os dois motores de tração do robô;
- Resolveu-se também retirar o sensor IR do projeto, pois os sensores ultrassônicos já estavam conseguindo visualizar todos os obstáculos;
- O código também sofreu leves alterações para aperfeiçoamento que pode ser visto mais a baixo e também na pasta no Github "MiniGramaFinal_PC4"

A figura abaixo mostra um esquemático da circuitaria do MiniGrama desenhado no Fritzing.



CÓDIGO

Getting Started | main.c | main.c

```
#include <msp430g2553.h>



//Motores
#define MOTORA1 BIT0 // P1
#define MOTORA2 BIT7 // P1

#define MOTORB1 BIT6 // P1
#define MOTORB2 BIT2 // P1
//

#define MOTORCORTE BIT2 //P2 // LIGADO EM NIVEL LÓGICO BAIXO
#define LIGAMOTORA1 (P1OUT |= MOTORA1)
#define DESLIGAMOTORA1 (P1OUT &= ~MOTORA1)
#define LIGAMOTORA2 (P1OUT |= MOTORA2)
#define DESLIGAMOTORA2 (P1OUT &= ~MOTORA2)
#define LIGAMOTORB1 (P1OUT |= MOTORB1)
#define DESLIGAMOTORB1 (P1OUT &= ~MOTORB1)
#define LIGAMOTORB2 (P1OUT |= MOTORB2)
#define DESLIGAMOTORB2 (P1OUT &= ~MOTORB2)
#define LIGAMOTORCORTE (P2OUT |= MOTORCORTE)
#define DESLIGAMOTORCORTE (P2OUT &= ~MOTORCORTE)

//Seletora do multiplexador
#define SEL1 BIT4
#define SEL0 BIT3

//Sensores ultrasssonico
#define triggerC BIT1
#define triggerD BIT2
#define triggerE BIT4
#define triggerB BIT0 ///P2
#define echo BIT5

```

Getting Started | main.c | main.c

```

//LEDs para teste
#define LED0 BIT0
#define LED1 BIT6
#define LED2 BIT5

//Direções/
#define CENTRO 0
#define DIREITA 1
#define ESQUERDA 2
#define EMBAIXO 3


int miliseconds;

long sensor;


typedef struct sensoresUltrassonicos
  {
    int Medicao[4];// Medições para entrada do filtro média móvel
    int valorFinal; // Resultado do filtro média móvel

  }sensoresUltrassonicos;


//Procedimentos
void filtro_Media_movel(sensoresUltrassonicos *sensor);
void viraParaDireita();
void viraParaEsquerda();
void ParaTudo();
void Reiniciar();
void ViraParaTras();
void SegueCortandoGrama();

```

Getting Started | main.c | main.c

```
69
70 void InicizalizaPortas();
71 //Funções
72 int MedeFrente();
73 int MedeDireita();
74 int MedeEsquerda();
75 int MedeEmbaixo();
76
77
78
79
80 int main(void)
81 {
82     InicizalizaPortas();
83     __delay_cycles(10000000);              // for 10us
84     LIGAMOTORA1;
85     DESLIGAMOTORA2;
86     LIGAMOTORB1;
87     //DESLIGAMOTORB2;
88     LIGAMOTORCORTE;
89     volatile int Frente = 0 , Direita = 0 ,Esquerda = 0 , Embaixo = 0;
90     while(1)
91     {
92
93         Frente = MedeFrente();
94         if(Frente < 30)
95         {
96             Direita = MedeDireita();
97             Esquerda =  MedeEsquerda();
98             if(Direita > 30)
99                 viraParaDireita();
100            else if(Esquerda > 30)
101                viraParaEsquerda();
102            else
103            {
104                ViraParaTras();
```

Getting Started | main.c | main.c

```
103            {
104                ViraParaTras();
105                Direita = MedeDireita();
106                Esquerda =  MedeEsquerda();
107                if(Direita > 30)
108                    viraParaDireita();
109                else if(Esquerda > 30)
110                    viraParaEsquerda();
111                else
112                    ParaTudo();
113
114            }
115        }
116
117        else
118        {
119            SegueCortandoGrama();
120
121        }
122
123
124
125
126    }
127
128
129 }
130
131 void viraParaDireita()
132 {
133     DESLIGAMOTORB1;
134     DESLIGAMOTORB2;
135     LIGAMOTORA1;
136     DESLIGAMOTORA2;
137     __delay_cycles(3000000);
138     SegueCortandoGrama();
```

Getting Started | main.c | main.c

```
141
142 void viraParaEsquerda()
143 {
144
145     DESLIGAMOTORA1;
146     DESLIGAMOTORA2;
147     LIGAMOTORB1;
148     DESLIGAMOTORB2;
149     __delay_cycles(3000000);
150     SegueCortandoGrama();
151
152
153
154 }
155 void ParaTudo()
156 {
157     DESLIGAMOTORA1;
158     DESLIGAMOTORA2;
159     DESLIGAMOTORB1;
160     DESLIGAMOTORB2;
161     DESLIGAMOTORCORTE;
162
163
164 }
165 void Reiniciar()
166 {
167     LIGAMOTORA1;
168     DESLIGAMOTORA2;
169     LIGAMOTORB1;
170     DESLIGAMOTORB2;
171     LIGAMOTORCORTE;
172
173 }
```

Getting Started | main.c | main.c

```
173 }
174 void ViraParaTras()
175 {
176     DESLIGAMOTORA1;
177     DESLIGAMOTORB1;
178     LIGAMOTORB2;
179     LIGAMOTORA2;
180     __delay_cycles(3000000);
181     SegueCortandoGrama();
182
183 }
184
185 int MedeFrente()
186 {
187     int nroMedicao = 0;
188     sensoresUltrassonicos distancia = {.Medicao = {0,0,0,0}, .valorFinal = 0};
189
190     P2OUT &= ~SEL1;
191     P1OUT &= ~SEL0;
192
193     while(nroMedicao <= 3)
194     {
195       __delay_cycles(10000);              // for 10us
196       P1IE &= ~0x01;            // disable interupt
197       P1DIR |= triggerC;          // trigger pin as output
198       P1OUT |= triggerC;          // generate pulse
199       __delay_cycles(10);             // for 10us
200       P1OUT &= ~triggerC;               // stop pulse
201       P1DIR &= ~echo;         // make pin P1.2 input (ECHO)
202       P1IFG = 0x00;                   // clear flag just in case anything happened before
203       P1IE |= echo;           // enable interupt on ECHO pin
204       P1IES &= ~echo;         // rising edge on ECHO pin
205     distancia.Medicao[nroMedicao] = sensor/58;
206     // Send_String("Distancia Centro:");
207    // Send_Int(distancia[CENTRO].valorFinal);
208     //Send_String("\n");
```

Getting Started | main.c | main.c

```
209
210     nroMedicao++;
211
212 }
213     filtro_Media_movel(&distancia);
214     return distancia.valorFinal;
215 }
216
217 int MedeDireita()
218 {
219     int nroMedicao = 0;
220     sensoresUltrassonicos distancia = {.Medicao = {0,0,0,0}, .valorFinal = 0};
221
222     P2OUT &= ~SEL1;
223     P1OUT |= SEL0;
224
225     while(nroMedicao <= 3)
226     {
227         __delay_cycles(10000);              // for 10us
228
229       P1IE &= ~0x01;        // disable interupt
230       P1DIR |= triggerD;          // trigger pin as output
231       P1OUT |= triggerD;          // generate pulse
232       __delay_cycles(10);             // for 10us
233       P1OUT &= ~triggerD;                 // stop pulse
234       P1DIR &= ~echo;         // make pin P1.2 input (ECHO)
235       P1IFG = 0x00;                       // clear flag just in case anything happened before
236       P1IE |= echo;           // enable interupt on ECHO pin
237       P1IES &= ~echo;         // rising edge on ECHO pin
238
239     distancia.Medicao[nroMedicao] = sensor/58;
240     //Send_String("Distancia Direita:");
241     //Send_Int(distancia[DIREITA].valorFinal);
242     //Send_String("\n");
243
```

Getting Started | main.c | main.c

```
243
244     nroMedicao++;
245
246     }
247     filtro_Media_movel(&distancia);
248     return distancia.valorFinal;
249
250 }
251
252 void SegueCortandoGrama()
253 {
254     LIGAMOTORA1;
255     DESLIGAMOTORA2;
256     LIGAMOTORB1;
257     DESLIGAMOTORB2;
258     LIGAMOTORCORTE;
259
260 }
261
262
263 int MedeEsquerda()
264 {
265
266     int nroMedicao = 0;
267     sensoresUltrassonicos distancia = {.Medicao = {0,0,0,0}, .valorFinal = 0};
268
269
270     P2OUT |= SEL1;
271     P1OUT &= ~SEL0;
272
```

Getting Started | main.c | main.c

```
273     while(nroMedicao <= 3)
274     {
275         __delay_cycles(10000);              // for 10us
276
277       P1IE &= ~0x01;       // disable interupt
278       P1DIR |= triggerD;          // trigger pin as output
279       P1OUT |= triggerD;          // generate pulse
280       __delay_cycles(10);             // for 10us
281       P1OUT &= ~triggerD;                 // stop pulse
282       P1DIR &= ~echo;         // make pin P1.2 input (ECHO)
283       P1IFG = 0x00;                     // clear flag just in case anything happened before
284       P1IE |= echo;           // enable interupt on ECHO pin
285       P1IES &= ~echo;         // rising edge on ECHO pin
286
287     distancia.Medicao[nroMedicao] = sensor/58;
288     //Send_String("Distancia Direita:");
289     //Send_Int(distancia[DIREITA].valorFinal);
290     //Send_String("\n");
291
292     nroMedicao++;
293
294     }
295     filtro_Media_movel(&distancia);
296     return distancia.valorFinal;
297
298
299 }
300 int MedeEmbaixo()
301 {
302     int nroMedicao = 0;
303     sensoresUltrassonicos distancia = {.Medicao = {0,0,0,0}, .valorFinal = 0};
304
305     P2OUT |= SEL1;
306     P1OUT |= SEL0;
```

Getting Started | main.c | main.c

```
307
308     while(nroMedicao <= 3)
309     {
310         __delay_cycles(10000);              // for 10us
311
312       P1IE &= ~0x01;       // disable interupt
313       P1DIR |= triggerD;          // trigger pin as output
314       P1OUT |= triggerD;          // generate pulse
315       __delay_cycles(10);             // for 10us
316       P1OUT &= ~triggerD;                 // stop pulse
317       P1DIR &= ~echo;         // make pin P1.2 input (ECHO)
318       P1IFG = 0x00;                     // clear flag just in case anything happened before
319       P1IE |= echo;           // enable interupt on ECHO pin
320       P1IES &= ~echo;         // rising edge on ECHO pin
321
322     distancia.Medicao[nroMedicao] = sensor/58;
323     //Send_String("Distancia Direita:");
324     //Send_Int(distancia[DIREITA].valorFinal);
325     //Send_String("\n");
326
327     nroMedicao++;
328
329     }
330     filtro_Media_movel(&distancia);
331     return distancia.valorFinal;
332
333 }
334
335
```

Getting Started main.c main.c

```
335
336
337
338 void filtro_Media_movel(sensoresUltrassonicos *sensor)
339 {
340     sensor->valorFinal= (sensor->Medicao[0]+ sensor->Medicao[1]+ sensor->Medicao[2] + sensor->Medicao[3])/4;
341 }
342
343 void InicizalizaPortas()
344 {
345     WDTCTL = WDTPW + WDTHOLD;
346     BCSCTL1 = CALBC1_1MHZ;
347     DCOCTL = CALDCO_1MHZ;
348
349     P2DIR |= SEL1;
350     P2OUT &= ~SEL1;
351     P1DIR |= SEL0;
352     P1OUT &= ~SEL0;
353
354 // Motores
355     P2SEL &= ~(BIT6 +BIT7);
356     P2SEL2 &= ~(BIT6 +BIT7);
357
358     P1DIR |= MOTORA1;
359     P1OUT &= ~MOTORA1;
360     P1DIR |= MOTORA2;
361     P1OUT &= ~MOTORA2;
362     P1DIR |= MOTORB1;
363     P1OUT &= ~MOTORB1;
364     P1DIR |= MOTORB2;
365     P1OUT &= ~MOTORB2;
366     P2DIR |= MOTORCORTE;
367     P2OUT &= ~MOTORCORTE;
368
369
370     CCTL0 = CCIE;                             // CCR0 interrupt enabled
```

Getting Started main.c main.c

```
366     P2DIR |= MOTORCORTE;
367     P2OUT &= ~MOTORCORTE;
368
369
370     CCTL0 = CCIE;                             // CCR0 interrupt enabled
371     CCR0 = 1000;                        // 1ms at 1mhz
372     TACTL = TASSEL_2 + MC_1;                  // SMCLK, upmode
373     CCTL0 = CCIE;                             // CCR0 interrupt enabled
374     _BIS_SR(GIE);
375
376 }
377 #pragma vector=PORT1_VECTOR
378 __interrupt void Port_1(void)
379 {
380     if(P1IFG&echo)  //is there interrupt pending?
381         {
382           if(!(P1IES&echo)) // is this the rising edge?
383           {
384             TACTL|=TACLR;   // clears timer A
385             miliseconds = 0;
386             P1IES |= echo;  //falling edge
387           }
388           else
389           {
390             sensor = (long)miliseconds*1000 + (long)TAR;    //calculating ECHO lenght
391
392           }
393     P1IFG &= ~echo;             //clear flag
394     }
395 }
396 #pragma vector=TIMER0_A0_VECTOR
397 __interrupt void Timer_A (void)
398 {
399   miliseconds++;
400 }
401
```