



UNIVERSIDADE DE AVEIRO
DEP. DE ENG. MECÂNICA

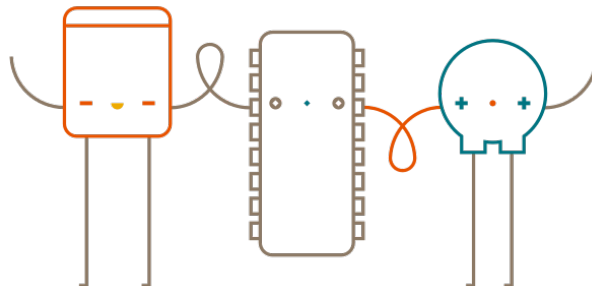
TECNOLOGIAS DE ACIONAMENTO E COMANDO

MESTRADO INTEGRADO EM ENGENHARIA MECÂNICA
PROJETO FINAL

Medição de distâncias através de um sensor de ultra-sons e um sensor industrial de medição precisa de distância.

Trabalho realizado por:

Joana Beatriz Carvalho Mota
Nº 72967



13 de Janeiro de 2018



Conteúdo

1	Introdução	2
2	Descrição do Projeto	2
3	Material Utilizado	2
3.1	ESP8266 12E com NodeMCU <i>firmware</i>	3
3.2	HC-SR04 Sensor de Ultra-sons	4
3.3	LCD 1602A V2.0	8
3.4	SENSIK DT20 Hi	12
3.5	LED's de Sinalização	18
4	Montagem Final	18
5	Programa Final	21
6	Conclusões	24



1 Introdução

Este trabalho tem como objetivo o aprofundamento do conhecimento em micro-controladores (*Microchip* ou ESP8266) e a sua comunicação com diversos periféricos/sensores. Neste trabalho, o micro-controlador será responsável tanto pelo controlo como pela monitorização dos dados obtidos pelos periféricos e o protocolo de comunicação utilizado entre estes e o micro-controlador variará consoante os mesmos, podendo ser do tipo SPI, I2C, Modbus/Rs485.

2 Descrição do Projeto

De modo a cumprir todos os objetivos foi então utilizado um micro-controlador para controlar a receção de dois sensores de medição de distâncias, um de ultra-sons e outro optoelectrónico. O micro-controlador será então responsável pela receção de ambos os sinais e pela interpretação e conversão dos mesmos em distância (mm). Visto que os sensores enviam sinais de tipos diferentes, sendo o sinal do ultra-sons digital e o do optoelectrónico analógico, a conversão em distância é realizada de forma diferente para cada um. Para a visualização das distâncias utilizou-se um LCD de 16x02 que comunica com o micro-controlador por I2C. Utilizaram-se também 3 LED's de cores diferentes, de modo a sinalizar os limites do sensor optoelectrónico.

Com este projeto é então possível verificar e comparar a exatidão de ambos os sensores, visto que o sensor optoelectrónico é um sensor industrial de grande precisão e o de ultra-sons é um sensor relativamente barato e com uma precisão de aproximadamente 3 mm.

Para a programação do micro-controlador foi utilizado o *Software open-source Arduino IDE*.

3 Material Utilizado

Utilizou-se neste projeto um ESP8266, um sensor de ultra-sons HC-SR04, um sensor optoelectrónico da *SICK*, um LCD de 16x02 com um módulo de interface I2C, 3 LED's e *jumpers* para realizar as ligações. São de seguida enumerados os materiais utilizados, juntamente com as suas características e modo de funcionamento.



3.1 ESP8266 12E com NodeMCU *firmware*

Este micro-controlador tem um consumo de energia consideravelmente baixo, muito utilizado para dispositivos móveis.

- **Características:**

O ESP8266 é composto por 15 pinos com acesso a GPIO's, SPI, I2C, UART, ADC e pinos de alimentação, tendo uma tensão operacional de 3.3V-3.6V. Tanto a alimentação (5V) do dispositivo como a programação é feita por um cabo USB - micro USB. O *datasheet* com todas as características do ESP-12E encontra-se nas referências [5].

A figura 1 é uma representação deste micro-controlador e dos seus pinos.

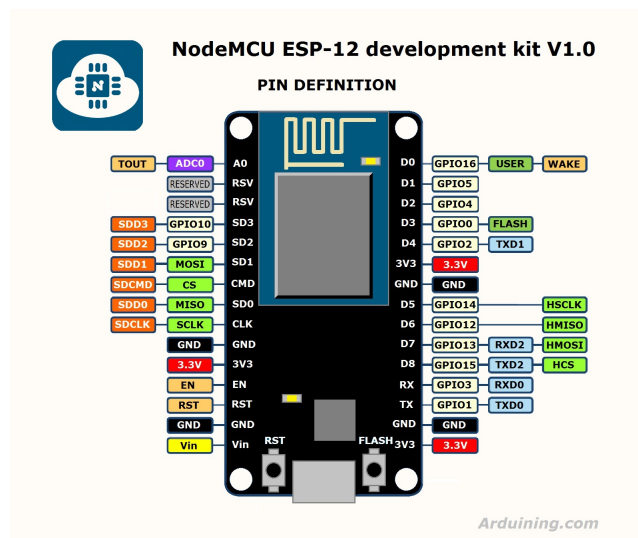


Figura 1: Pinos do ESP8266 12E.



3.2 HC-SR04 Sensor de Ultra-sons

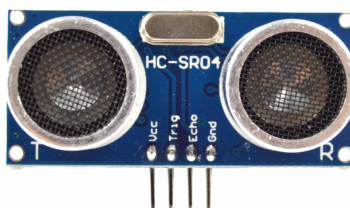


Figura 2: Sensor de Ultra-sons.

- **Características:**

Este sensor possibilita uma medição sem contacto, entre 2cm e 4m e com uma precisão de 3mm. Os módulos incluem um transmissor (*Trigger*), um recetor *Echo* e um circuito de controlo ultrassónico.

- Tensão Operacional : DC 5V
- Corrente Operacional : 15mA
- Ângulo de Medição : 15°
- *Trigger Input Signal* : Sinal TTL de 10 μ s
- *Echo Output Signal* : Sinal TTL com comprimento proporcional á distância.

É possível visualizar outras características no *datasheet* apresentado nas referências [3].



- **Funcionamento:**

Depois de alimentado, o sensor ativa o pino *Trigger* durante $10\mu\text{s}$, como um sinal de *Start*. Este emite 8 ciclos de uma onda sonora de 40kHz e espera pelo sinal de retorno caso este detete algum objeto. Depois da emissão da onda, o pino *Echo* é ativado até ao retorno de sinal. O diagrama dos sinais e o modo de funcionamento são representados nas figuras seguintes.

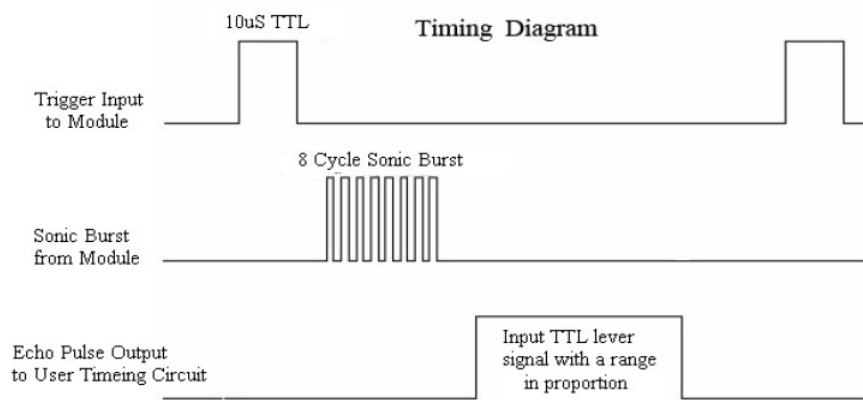


Figura 3: Sinal HC-SR04

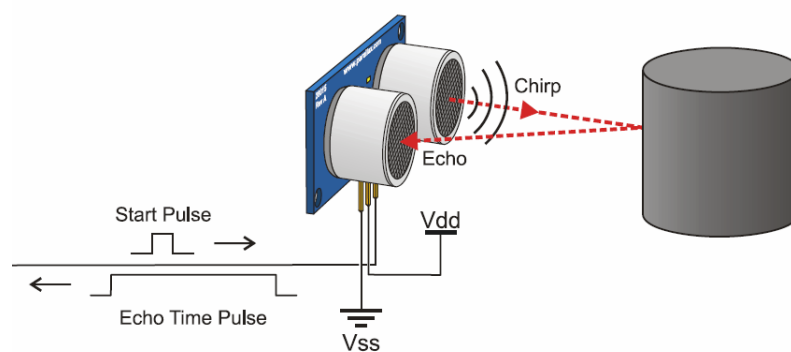


Figura 4: Esquema do funcionamento do sensor de ultra-sons



Tendo a duração do sinal do *Echo* em estado ativo, é possível determinar a distância a partir da velocidade do som (340m/s) e da seguinte equação:

$$Distancia = \frac{Duracao \times 0.34}{2}$$

De modo a comprovar o correto funcionamento do dispositivo foram visualizados dois sinais do sensor, obtidos por um osciloscópio, para duas distâncias diferentes. A figura 5 apresenta o sinal adquirido para uma distância de 10cm e a figura 6 para uma distância de 20cm, ambos com uma base de tempo de $500\mu s$.

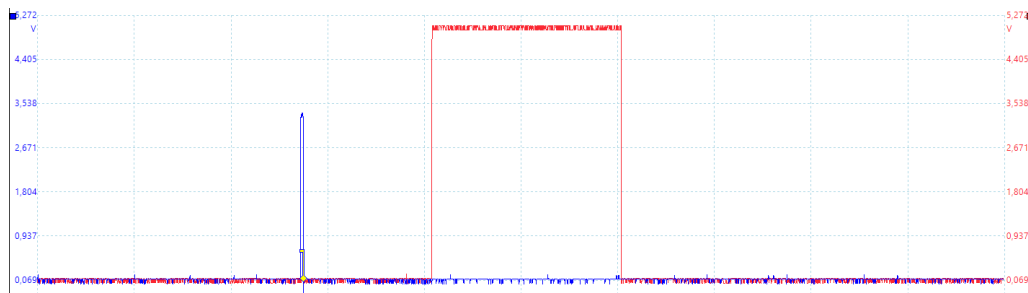


Figura 5: Sinal do sensor de ultra-sons para uma distância de 10cm.

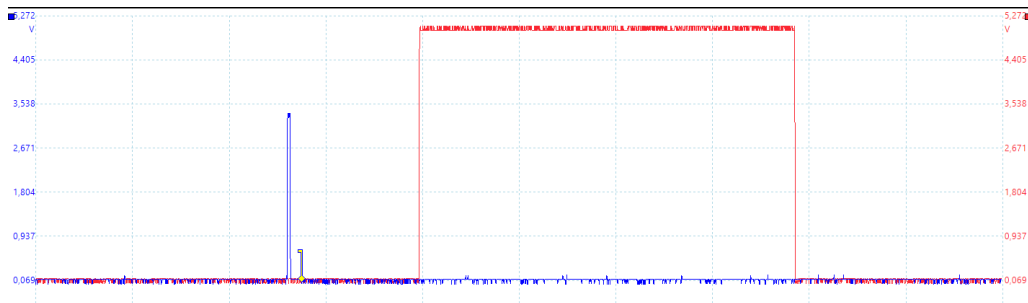


Figura 6: Sinal do sensor de ultra-sons para uma distância de 20cm.



- **Circuito:**

Tendo que este sensor necessita de ser alimentado com 5V, o pino Vcc deste terá que ser conectado com o pino VU do ESP8266 que envia 5V. Todas as ligações do sensor de ultra-sons e o ESP8266 são apresentadas na figura 7.

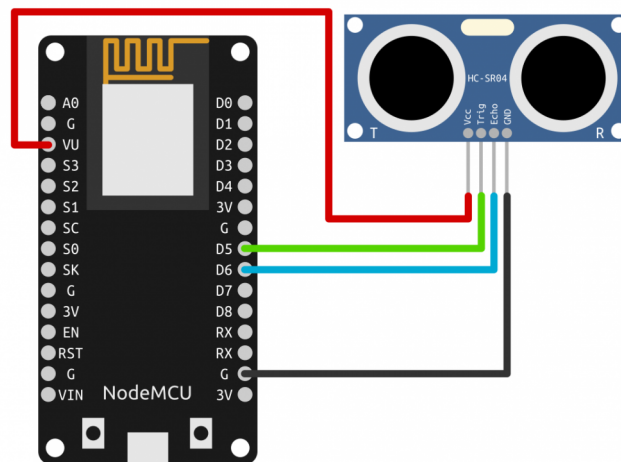


Figura 7: Circuito HC-SR04 - ESP8266

- **Código:**

A secção do programa responsável pela receção do sinal e do cálculo da distância encontra-se no código seguinte.

```

1 //-----HC-SR04-----
2 // Clears the trigPin
3 digitalWrite(trigPin , LOW); //PULSE ---|---|---
4 delayMicroseconds(2);
5 // Sets the trigPin on HIGH state for 10 micro seconds
6 digitalWrite(trigPin , HIGH);
7 delayMicroseconds(10);
8 digitalWrite(trigPin , LOW);
9
10 // Reads the echoPin, returns the sound wave travel time
    in microseconds
11 duration = pulseIn(echoPin , HIGH);
12 // Calculating the distance
13 distance = duration*0.34/2;

```




3.3 LCD 1602A V2.0

LCD's de apresentação de caracteres são já muito utilizados, sendo possível encontrá-los em máquinas de café, brinquedos para crianças, ponteiros laser, entre muitos outros equipamentos. Um dispositivo como este é muito útil para a exibição de simples informação, como neste caso a distância obtida por dois sensores.

O *display LCD* utilizado é muito compacto, versátil e capaz de representar até 32 caracteres.

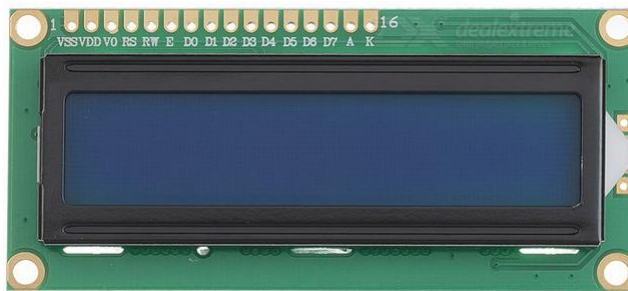


Figura 8: LCD 1602A V2.0

- **Características:**

- Modo de *Display* : STN, BLUB
- Formato do *Display* : 16 caracteres por 2 linhas
- *Input Data* : Interface de 4bits ou 8bits
- Tensão Operacional : 5V
- *Backlight* (lateral) : LED (azul)

- **Funcionamento:**

Um LCD, como o nome diz é um *liquid crystal display*, composto por uma substância que não é nem líquida nem sólida, sendo um intermédio entre os dois. Isto porque o *liquid crystal* se encontra num estado em que as suas moléculas tendem a manter a mesma



orientação, como as moléculas de um sólido, mas também se movem para diferentes posições, como as moléculas de um líquido. Ao aplicar uma carga elétrica às moléculas de *liquid crystal*, estas distorcem e ao endireitarem-se alteram o ângulo da luz que as atravessa, aparecendo umas zonas mais escuras e outras mais claras. Todavia, esta substância não emite nenhuma luz sendo por isso necessário uma luz externa, como por exemplo um LED [6].

Para a construção de um LCD são necessárias várias camadas, sendo uma delas uma grelha utilizada para aplicar uma carga a um pixel específico do *display*. De modo a programar esses píxeis, são enviados para o módulo 8 bits de dados que ativam os pinos desde o D1 ao D8. A ativação do pino RS indica ao módulo se a informação enviada são dados ou comandos. Sempre que é necessário apresentar um carácter é necessário ativar os pinos respetivos, para isso foram então utilizadas bibliotecas. Porém, mesmo com estas bibliotecas não existe uma simplificação das ligações do LCD ao micro-controlador, apenas do código em si, e sendo o ESP8266 composto por um reduzido número de entradas, as ligações tornam-se mais confusas e complicadas. Para solucionar este problema é então utilizado um conversor I2C (MCP23008).

O protocolo “*Inter Integrated Circuit*” - I2C foi proposto pela Philips Electronics para permitir a troca de dados entre os seus circuitos integrados. Este protocolo utiliza apenas dois condutores para ligar dois ou mais dispositivos: um condutor é usado para transmitir um sinal de relógio “*Serial Clock*” - SCL; e o outro condutor permite aos equipamentos enviar ou receber dados “*Serial Data*” - SDA (bidirecional). Desta forma é possível comunicar com o micro-controlador apenas por 2 pinos.

- **Circuito:**

Todas as ligações realizadas entre o LCD e o ESP8266 são apresentadas na figura 9 e na seguinte tabela.

LCD	ESP8266
GND	G
5V	VU (5V)
SDA	D2
SCL	D1

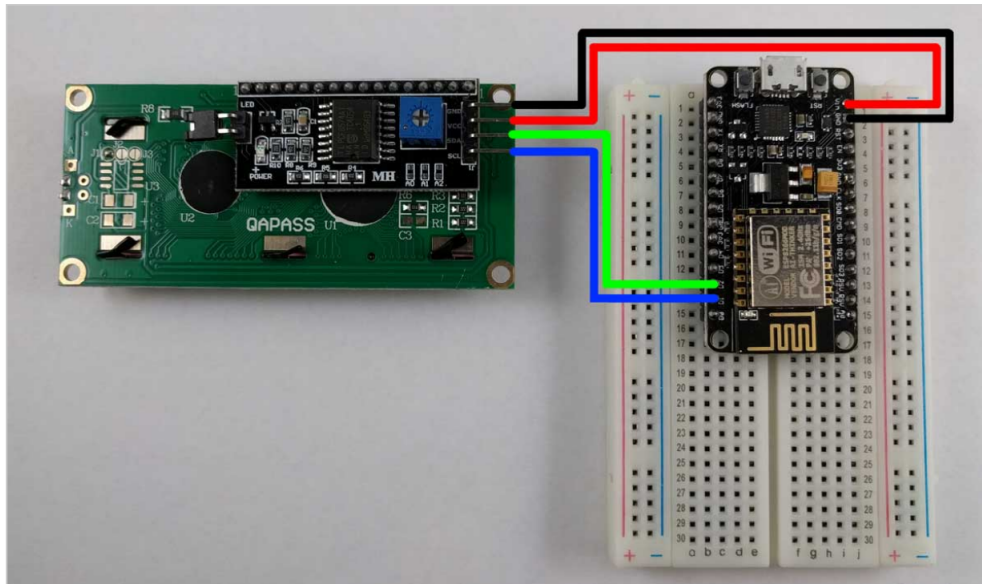


Figura 9: Circuito LCD 1602 - ESP8266

- **Código:**

As bibliotecas utilizadas para a conversão do sinal e apresentação no LCD são as apresentadas no código seguinte, estas são as adequadas para uma conversão I2C que utiliza um expansor MCP23008. A descrição da biblioteca *Wire* encontra-se no site oficial do Arduino, cujo *link* se encontra nas referências [1]. A biblioteca *Adafruit_LiquidCrystal* encontra-se *online* no repositório *GitHub* juntamente com alguns exemplos [2].

De modo a comunicar com o dispositivo desejado, visto que é possível comunicar com vários dispositivos ao mesmo tempo com I2C, é necessário codificar o endereço respetivo, apresentado igualmente no código seguinte (0x20). O endereço deste dispositivo foi obtido através de um simples programa que deteta a existência de um dispositivo e obtém o seu endereço, cujo *link* se encontra igualmente nas referências .



```
1 #include "Wire.h"
2 #include "Adafruit_LiquidCrystal.h"
3
4 // Connect via i2c, address--0x20-->obtained with i2c_Scanner
5 Adafruit_LiquidCrystal lcd(0x20);
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

A secção do programa responsável pela apresentação das distâncias dos 2 sensores encontra-se no código seguinte.

```
1 //-----LCD-----
2 lcd.setCursor(3,1);
3 lcd.print(" ");
4 lcd.setCursor(14,1);
5 lcd.print(" ");
6 lcd.setCursor(0,1);
7 lcd.print(distance);
8 lcd.print("mm");
9 lcd.setCursor(11,1);
10 lcd.print(DistanceSENSICK);
11 lcd.print("mm");
12
13 delay(500);
```



3.4 SENSICK DT20 Hi

O sensor SENSICK DT20 Hi é um dispositivo optoelectrónico que permite obter distâncias entre objetos via ótica e sem contacto. O emissor é um laser de cor vermelha com comprimento de onda de 655nm, que permite uma boa leitura da distância, independentemente da rugosidade do objeto onde o laser incide.



Figura 10: SENSICK DT20 HI

- **Características:**

- Alcance da Medição : 100mm - 1,000mm
- Resolução : 1,000 μm
- Linearidade : $6 \pm \text{mm}$
- Velocidade de Reação : 2,5ms/10 ms/40 ms, consoante a configuração realizada no menu do sensor (*fast/medium/slow*)
- Tensão Operacional : 10V - 30V
- Saída Analógica : 4mA - 20mA, com uma resolução de 12bits

Outras características podem ser encontradas no *Data Sheet* deste sensor [4].

- **Funcionamento:**

O díodo existente no sensor emite um pulso laser, que é refletido pelo objeto a ser detetado. O sensor mede e avalia o tempo que o sinal demora desde o díodo até ao objeto e a voltar. Posteriormente, a distância é calculada a partir desse tempo e da velocidade da luz. A representação desse fenómeno é apresentado na figura 11.

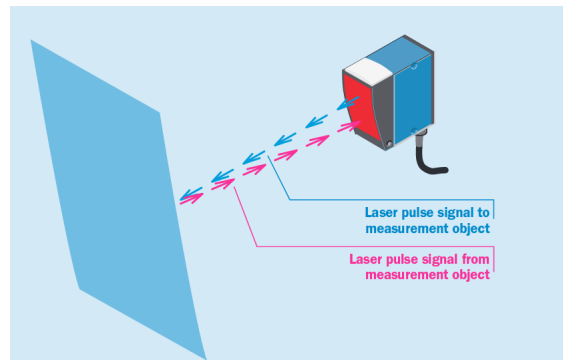


Figura 11: Funcionamento do sensor SENSICK DT20 Hi

Este sensor da SICK tem duas saídas de sinal, uma analógica e uma de comutação, e tem ainda uma entrada multifunções, como é possível visualizar na figura 12. Para este projeto foi usada a saída analógica que se comporta como fonte de corrente. Apesar da gama de corrente vir já predefinida para a gama total do sensor, é possível configurá-la a partir do seu menu incorporado. Por se ter optado pelo uso de uma gama de 200mm a 700mm, a corrente irá variar entre 4mA e 20mA entre estas duas distâncias.

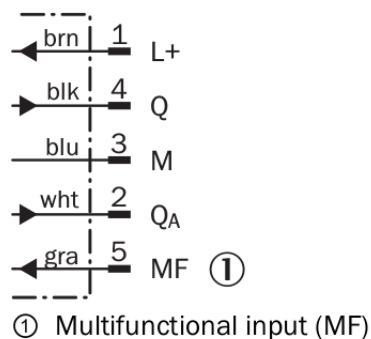


Figura 12: Diagrama das ligações do sensor SENSICK DT20 Hi

- **Circuito:**

Como a saída analógica do sensor age como uma fonte de corrente e a entrada analógica do ESP8266 faz uma leitura de tensão, foi necessário



converter a corrente da saída do sensor numa tensão proporcional à mesma, utilizando uma resistência em série. Foi usada neste projeto uma resistência de $100\ \Omega$, sendo que a tensão que entra no ESP será no máximo de 2V. Poderia ter sido utilizada uma resistência de até $160\ \Omega$, obtendo assim uma tensão de aproximadamente 3.3V, de modo aproveitar ainda mais a resolução do sinal enviado pelo sensor. De referir também, que seria ainda mais viável utilizar um *Arduino* e uma resistência de $250\ \Omega$ de modo a obter uma tensão de 5V, visto que este o possibilita.

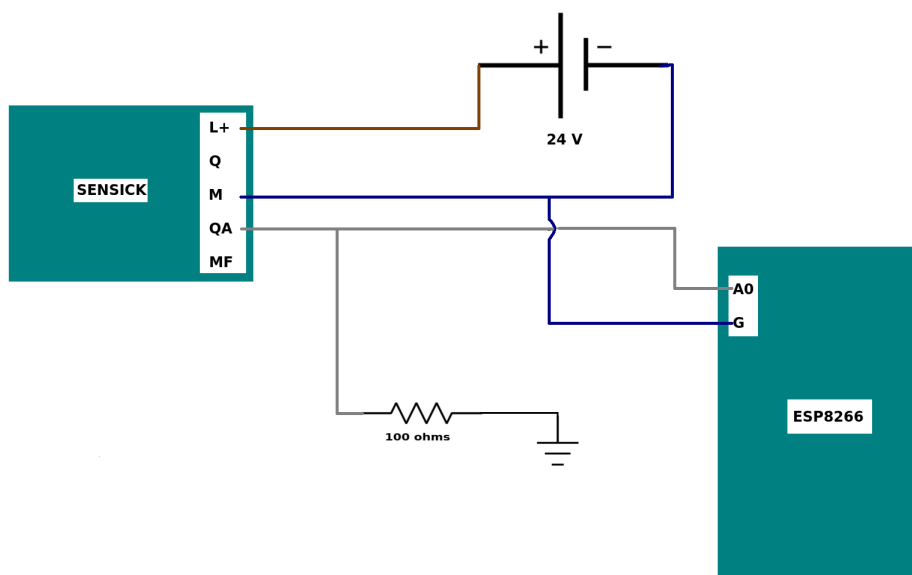


Figura 13: Circuito SENSICK - ESP8266

- **Aquisição de dados e conversão:**

Depois de tudo corretamente conectado e funcional, inicializou-se de seguida a aquisição de dados. Visto que se utilizou uma resistência de $100\ \Omega$ e se limitou o sensor através do menu do mesmo, a uma gama de leitura entre os 200mm e os 700mm, os valores de tensão irão variar aproximadamente entre os 0V e 2V. Recolheu-se 20 valores da queda de tensão na resistência e da leitura A0 do ESP, na gama de leitura predefinida. Os valores da leitura são apresentados na tabela seguinte.



Distância Sensor (mm)	Tensão (V)	Leitura A0 (ESP)	Distância (mm)
200	1.97	589	201.26
225	1.89	566	225.89
250	1.81	543	250.52
275	1.73	521	275.14
300	1.65	497	299.77
325	1.57	474	324.40
350	1.49	450	349.03
375	1.41	428	373.66
400	1.34	405	398.29
425	1.26	380	426.13
450	1.18	356	451.83
475	1.10	333	475.38
500	1.02	310	500.01
525	0.94	286	525.71
550	0.86	264	549.27
575	0.78	240	574.97
600	0.71	216	599.60
625	0.63	194	625.30
650	0.55	170	649.92
675	0.47	147	674.62
700	0.30	122	702.39

A partir dos valores da leitura do sinal e da distância apresentada no *display* do sensor, foi determinada a equação da reta que relaciona ambos, cujo gráfico se encontra apresentado na figura 14. Com base neste é possível demonstrar a linearidade desta relação. Utilizou-se portanto a seguinte equação para determinar a distância a que se encontram os objetos em função da leitura, cujos os valores se encontram na tabela apresentada anteriormente.

$$Dinstancia = -1.0708 \times Leitura + 831.96$$

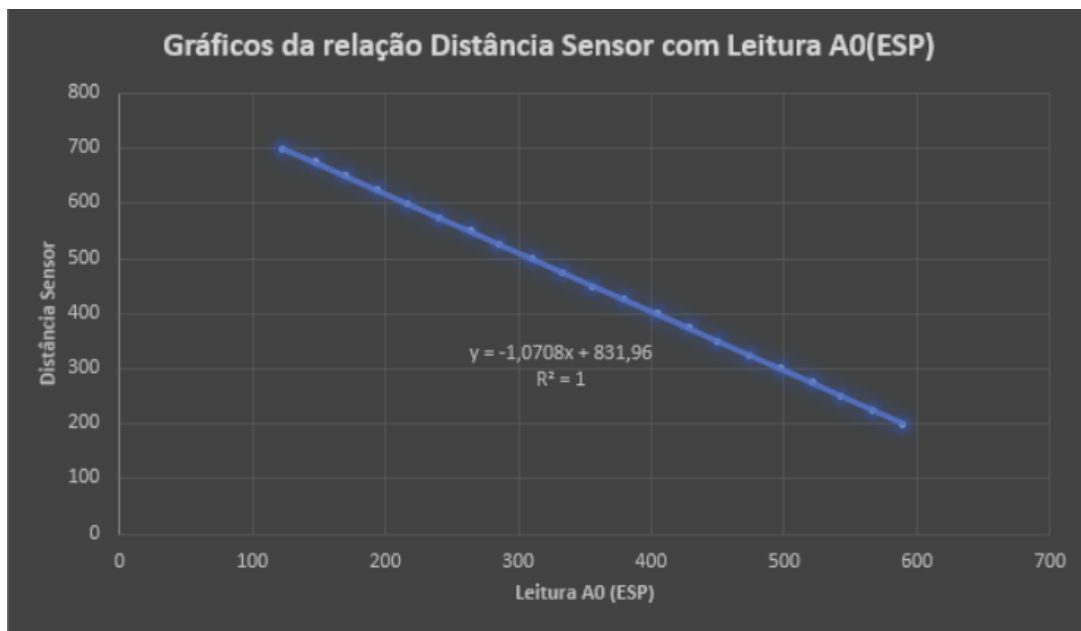


Figura 14: Gráfico da relação entre a distância apresentada no sensor e a leitura A0

- **Resolução do Sensor:**

A saída analógica deste sensor tem uma resolução de 12 bits, e visto que a entrada analógica do ESP8266 (A0) interpreta apenas 10 bits, a sua resolução é diminuída. Por esta razão, os valores obtidos na leitura através do ESP8266 terão uma discrepância do valor real enviado pelo sensor, como é possível visualizar na tabela. Seria portanto mais viável utilizar um hardware capaz de interpretar os 12bits enviados pelo sensor.

Para além disto, será necessário reduzir frequências parasitas na tensão medida, para tal será implementado um filtro passa-baixo. A representação do circuito utilizando este filtro é apresentada na figura 20.

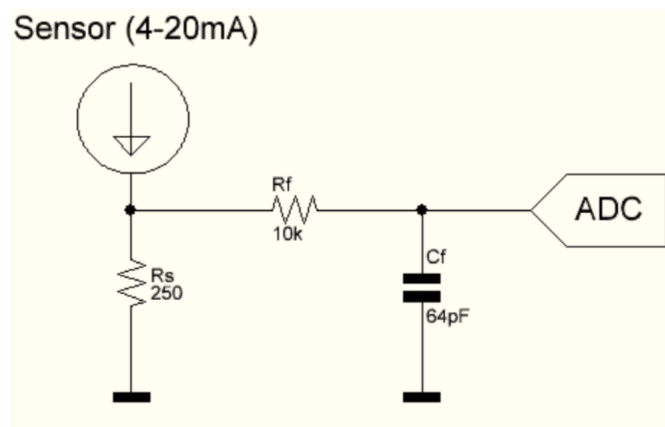


Figura 15: Filtro Passa-Baixo

- **Código:**

De seguida é apresentada uma parte do código principal, onde é lido o valor do sinal analógico do sensor. A variável *DistanceSENSICK* é do tipo *float* de modo a obter mais duas casa decimais a seguir à virgula, para uma melhor análise dos valores.

```
1 //-----SENSIK-----
2 // read the analog in value:
3 sensorValue = analogRead(analogInPin);
4
5 // print the results to the Serial Monitor:
6 Serial.print("sensor = ");
7 Serial.print(sensorValue);
8
9 DistanceSENSICK=-1.0708*sensorValue+831.96;
10
11 Serial.println(DistanceSENSICK);
12
13 // wait 2 milliseconds before the next loop for the analog
14 // converter to settle after the last reading:
15 delay(2);
```



3.5 LED's de Sinalização

Utilizaram-se 3 LED's com o intuito de sinalizar os limites do sensor SENSICK. Sinalizou-se as distâncias inferiores a 200mm com um LED de cor amarela, o intervalo entre os 200 e os 700mm com um LED de cor verde, e valores de distâncias superiores a 700mm foram sinalizados com um LED de cor vermelha. Na ligação destes com o ESP8266 foram todos acompanhados por uma resistência em série de 1 K Ω .

4 Montagem Final

São apresentadas de seguida algumas imagens que demonstram a montagem e funcionamento de todo o projeto.

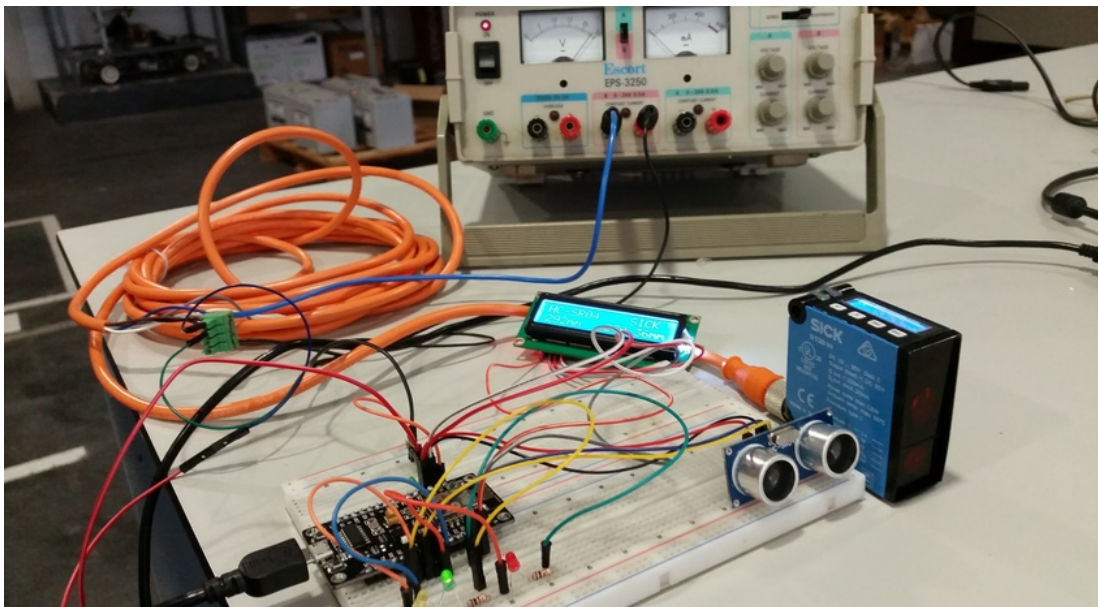


Figura 16: Circuito com todos os componentes.

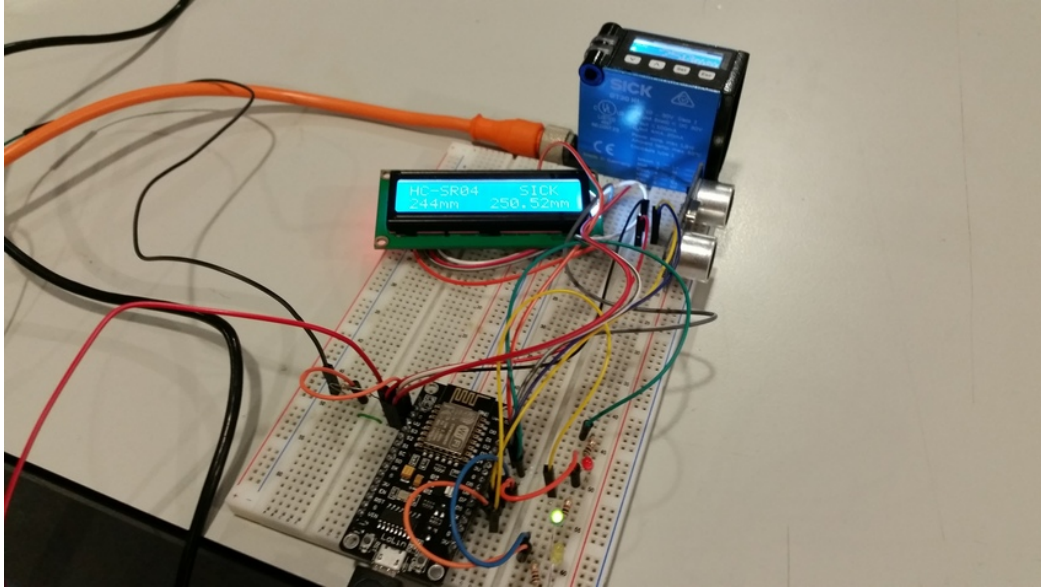


Figura 17: Circuito de ligação do ESP8266 com os vários periféricos.

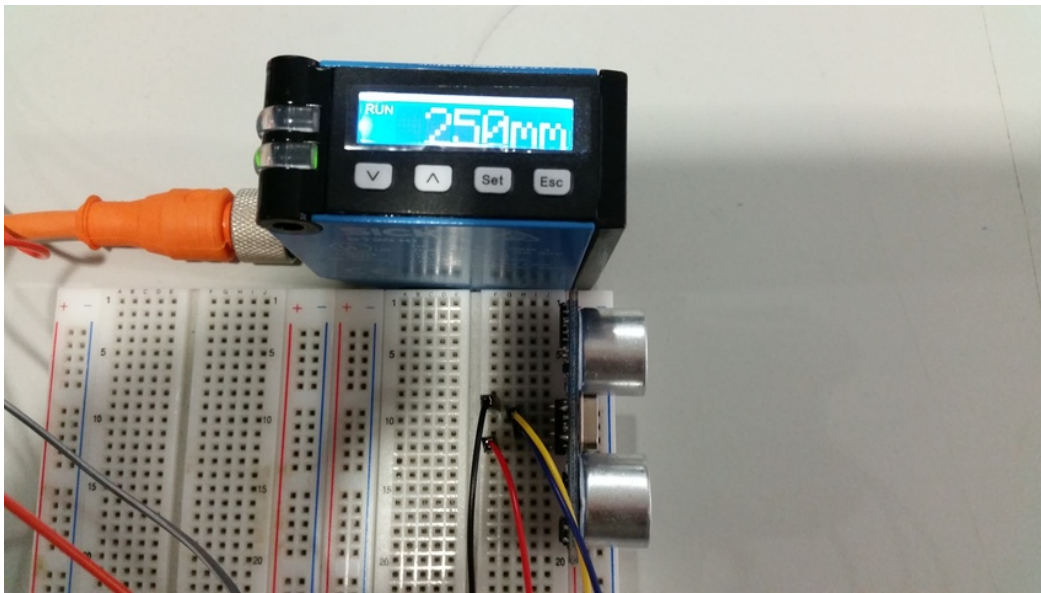


Figura 18: Sensor de ultra-sons e sensor optoelectrónico.



Figura 19: *Display* das leituras obtidas pelos 2 sensores e comparação entre a leitura obtida do sensor optoelectrónico e a apresentada por este.

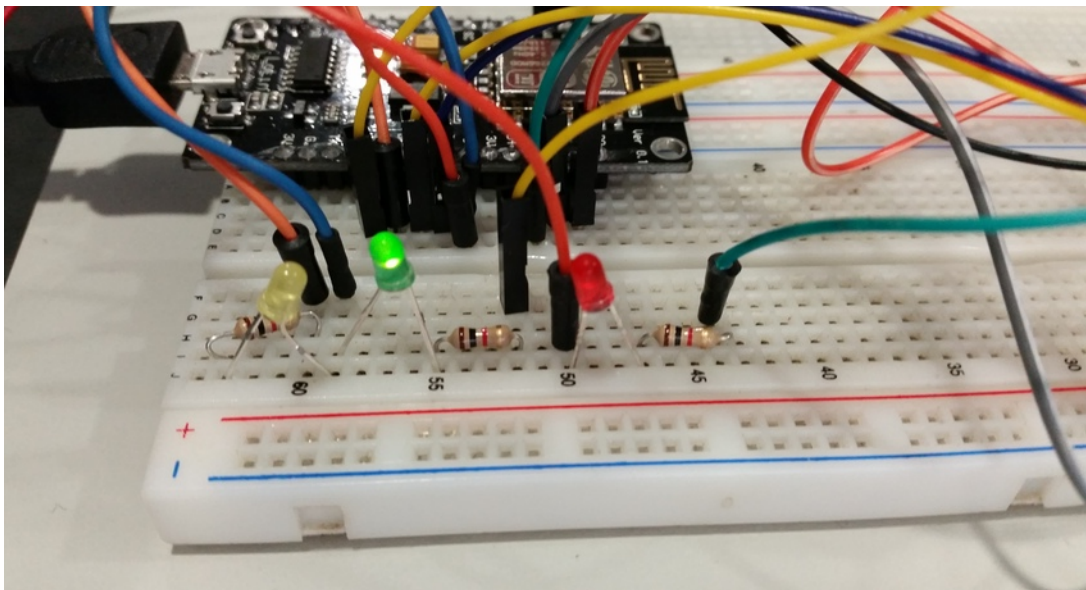


Figura 20: LED's de sinalização.



5 Programa Final

```
1 /*
2  * Trabalho Pratico de TAC
3  * obtain information from a sensor with high precision from
4  * SENSIK
5  * and a ultrasonic sensor HC-SR04.
6  * Output information in a LCD display
7  */
8
9 /*Include the library code for the LCD
10 *
11 * The circuit:
12 * 5V to VU--5V
13 * GND to G
14 * SCL to D1
15 * SDA to D2
16 */
17 #include "Wire.h"
18 #include "Adafruit_LiquidCrystal.h"
19
20 // Connect via i2c, adress--0x20-->obtained with i2c_Scanner
21 Adafruit_LiquidCrystal lcd(0x20);
22
23
24 /*for HC-SR04
25 *
26 * The circuit:
27 * Vcc to VU--5V
28 * GND to G
29 * Trig to D5
30 * Echo to D6
31 *
32 */
33 #define trigPin D5
34 #define echoPin D6
35 long duration, distance; //our beloved variables
36
37 //for SENSIK
38 // These constants won't change. They're used to give names to
39 // the pins used:
40 const int analogInPin = A0; // Analog input pin that the
    potentiometer is attached to
```



```
41 int sensorValue = 0;          // value read from the pot
42 int outputValue = 0;         // value output to the PWM (analog
                                out)
43 long DistanceSENSICK;
44
45 //LED's control
46 #define yellow200 D7
47 #define green200_700 D8
48 #define red700 D3
49
50
51 void setup() {
52     // configurate all the sensors
53
54     // initialize serial communications at 9600 bps:
55     Serial.begin (9600);
56
57
58     //—————HC-SR04—————
59     pinMode(trigPin , OUTPUT);
60     pinMode(echoPin , INPUT);
61
62     //—————LCD—————
63     // set up the LCD's number of rows and columns:
64     lcd.begin(16, 2);
65     // Print a message to the LCD.
66     lcd.setBacklight(HIGH);
67     lcd.print("Hello , Joana!");
68     delay(2000);
69     lcd.clear();
70     lcd.print("HC-SR04  SICK");
71     pinMode(yellow200 , OUTPUT);
72     pinMode(green200_700 , OUTPUT);
73     pinMode(red700 , OUTPUT);
74
75 }
76
77 void loop() {
78
79     //—————SENSIK—————
80     // read the analog in value:
81     sensorValue = analogRead(analogInPin);
82
83     // print the results to the Serial Monitor:
84     Serial.print("sensor = ");
```



```
85 Serial.print(sensorValue);
86
87 DistanceSENSICK=-1.0708*sensorValue+831.96;
88
89 Serial.println(DistanceSENSICK);
90
91 // wait 2 milliseconds before the next loop for the analog-to-
92 // digital
93 // converter to settle after the last reading:
94 delay(2);
95
96 //—————HC-SR04—————
97 // Clears the trigPin
98 digitalWrite(trigPin, LOW); //PULSE ---|---|---
99 delayMicroseconds(2);
100 // Sets the trigPin on HIGH state for 10 micro seconds
101 digitalWrite(trigPin, HIGH);
102 delayMicroseconds(10);
103 digitalWrite(trigPin, LOW);
104
105 // Reads the echoPin, returns the sound wave travel time in
106 // microseconds
107 duration = pulseIn(echoPin, HIGH);
108 // Calculating the distance
109 distance = duration*0.34/2;
110
111 Serial.print("Distance= ");
112 Serial.print(distance); //debug
113 Serial.print("mm\n");
114
115
116
117 //—————LCD—————
118 lcd.setCursor(3,1);
119 lcd.print(" ");
120 lcd.setCursor(14,1);
121 lcd.print(" ");
122 lcd.setCursor(0,1);
123 lcd.print(distance);
124 lcd.print("mm");
125 lcd.setCursor(11,1);
126 lcd.print(DistanceSENSICK);
127 lcd.print("mm");
```




```
128
129   delay(500);
130
131
132   if (DistanceSENSICK < 200)
133   {
134     digitalWrite(yellow200, HIGH);
135     digitalWrite(green200_700, LOW);
136     digitalWrite(red700, LOW);
137   }
138   else if (DistanceSENSICK >= 200 && DistanceSENSICK < 700)
139   {
140     digitalWrite(yellow200, LOW);
141     digitalWrite(green200_700, HIGH);
142     digitalWrite(red700, LOW);
143
144   }
145   else if (DistanceSENSICK > 700)
146   {
147     digitalWrite(yellow200, LOW);
148     digitalWrite(green200_700, LOW);
149     digitalWrite(red700, HIGH);
150   }
151
152 }
```

6 Conclusões

Ao longo deste relatório foram já mencionadas algumas vantagens e desvantagens da utilização destes dispositivos e dos protocolos de comunicação. No entanto, é necessário realçar a vantagem da utilização do protocolo I2C para estabelecer a comunicação com o LCD, evitando desta forma um complicado circuito que seria necessário utilizar para ativar os pinos responsáveis pela representação dos caracteres. É de realçar também que, de modo a obter leituras do sensor optoelectrónico mais precisos, será necessário utilizar um *hardware* capaz de interpretar os 12 bits do sinal analógico. Para além de melhorar a resolução do sinal, é imperativo eliminar o ruído do mesmo através da utilização de um filtro passa-baixo.



Referências

- [1] Arduino. Wire library. <https://www.arduino.cc/en/Reference/Wire>, 2017.
- [2] Tony DiCola. Adafruit_liquidcrystal. GitHub - https://github.com/adafruit/Adafruit_LiquidCrystal, 2017.
- [3] Elecfreaks. Ultrasonic ranging module hc - sr04. Technical report, Elecfreaks, 2017.
- [4] SICK. Dt20-n224b — dt20 hi displacement measurement sensors. Technical report, SICK Sensor Intelligence, 2017.
- [5] AI Thinker team. Esp - 12e wifi module. <http://www.kloppenborg.net/images/blog/esp8266/esp8266-esp12e-specs.pdf>, 2015.
- [6] Jeff Tyson. How lcds work. <https://electronics.howstuffworks.com/lcd.htm>, 2017.