



universidade de aveiro

# Inclinómetro planar de precisão para o Atlascar-2

Projeto de Automação Industrial – Relatório Final

Orientador- Prof. Vítor Santos

82660 – Armindo Silva

Aveiro, 16 junho 2017

# Índice

Índice .....	2
Lista de Figuras .....	3
1. Resumo .....	4
2. Introdução.....	4
3. Objetivos do projeto .....	4
4. Elementos do Projeto.....	5
4.1. Sensor SENSICK DT20 Hi .....	5
4.2. Microcontrolador Arduino Nano.....	6
4.3. LCD 16x2 .....	6
5. Implementação da Hardware .....	7
6. Aquisição de dados e conversão.....	9
7. Implementação do programa no Arduino nano .....	12
8. Montagem no veículo.....	14
9. Testes de funcionalidade no veículo.....	16
10. Comunicação por Bluetooth.....	20
11. Conclusões .....	21

# Lista de Figuras

<i>Figura 1-Sensor SENSICK DT20 Hi</i> .....	5
<i>Figura 2-Arduino Nano AT328</i> .....	6
<i>Figura 3-LCD 16x2</i> .....	6
<i>Figura 4-Resistência queda de tensão e filtro passa baixo</i> .....	7
<i>Figura 5- Esquema realizado em eagle</i> .....	8
<i>Figura 6-layout do PCB realizado em eagle</i> .....	8
<i>Figura 7-Dados recolhidos e valores de leitura de distância convertida</i> .....	9
<i>Figura 8-Valores de regressão para um dos quatro sensores</i> .....	9
<i>Figura 9-Gráficos da relação Distância Sensor com Leitura ADC</i> .....	11
<i>Figura 10-Ilustração da obtenção dos ângulos</i> .....	12
<i>Figura 11-Programa para a obtenção dos ângulos</i> .....	13
<i>Figura 12-Posição dos sensores traseiros</i> .....	14
<i>Figura 13-Posição dos sensores da frente</i> .....	14
<i>Figura 14-Caixa PVC com hardware, ligações realizadas e em função</i> .....	15
<i>Figura 15-Frequência de ocorrência de valores ADC</i> .....	17
<i>Figura 16-Frequência de ocorrência de valores ADC convertidos</i> .....	17
<i>Figura 17-Frequência de ocorrência de valores Pitch</i> .....	18
<i>Figura 18-Frequência de ocorrência de valores Roll</i> .....	18
<i>Figura 19-Frequência de ocorrência de valores Pitch com media de valores</i> .....	19
<i>Figura 20-Frequência de ocorrência de valores Roll com media de valores</i> .....	19
<i>Figura 21-Aplicação android realizada na MIT appinventor</i> .....	20
<i>Figura 22-Diagrama de blocos do programa na MIT appinventor</i> .....	20

# 1. Resumo

---

Este projeto consistiu em desenvolver um sistema que permite determinar a inclinação de um veículo quando este é sujeito a acelerações. Esta calculo é baseado na medição da distância de quatro pontos distintos do chassi em relação ao chão.

## 2. Introdução

---

Este trabalho consiste na realização de um inclinómetro que devolve informação sobre os ângulos de inclinação do carro quando este é sujeito a acelerações. Para esse efeito foram usados sensores de medição de distância. Estes medem a distância entre quatro pontos fixos do chassi do carro a uma referência que, neste caso, é a estrada. Quando o carro é sujeito a acelerações como, por exemplo, ao travar as forças aplicadas na estrutura do carro levam este a rebaixar a frente e levantar a traseira. Graças à diferença de distância medida em relação ao chão e informação sobre a distância entre sensores, pode ser, através de cálculo trigonométrico, calculado o ângulo de inclinação.

## 3. Objetivos do projeto

---

- Estudo da solução anterior desenvolvida para o ATLASCAR-1.
- Conceção geral do sistema e dos seus blocos constituintes.
- Desenvolvimento da unidade para interface elétrica entre sensores e sistema de aquisição.
- Programação do sistema de aquisição de dados e cálculo das orientações.
- Ensaios laboratoriais para teste e calibração do sistema de medição.
- Colocação, fixação e teste do sistema no carro.

## 4. Elementos do Projeto

---

### 4.1. Sensor SENSICK DT20 Hi

---

Para a obtenção das distâncias entre pontos fixos da estrutura do carro e o solo foi, para este projeto, usado o sensor SENSICK DT20 Hi. Este sensor 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, o que permite uma boa leitura da distância, independentemente da rugosidade do objeto de incidência do laser.

O sensor permite uma leitura de distância com uma gama de 50mm a 1000mm com resolução inferior a 1mm, linearidade de  $\pm 6\text{mm}$  e velocidade de reação de 2,5/ 10/ 40ms configurável no menu do sensor.

O sensor pode ser alimentado por uma fonte de tensão contínua com valores de tensão entre 10 e 30V. Tem duas saídas de sinal, uma analógica e uma de comutação, e tem uma entrada multifunções. Para este projeto foi usado a saída analógica que se comporta como fonte de corrente com uma gama de corrente que varia entre 4mA e 20mA. Esta gama de corrente vem predefinida para a gama total do sensor, no entanto pode ser alterada no menu. Como para este projeto a gama total do sensor não é necessária e o sensor esta montado a 450mm do solo optou-se por uma gama de 500mm, de 200mm a 700mm, o que corresponde, respetivamente, a uma variação na corrente de 4mA a 20mA.



Figura 1-Sensor SENSICK DT20 Hi

## 4.2. Microcontrolador Arduino Nano

---

Para este projeto usou-se um Arduino nano com microcontrolador AT328. O Arduino tem uma tensão de operação de 5V e pode ser alimentado por uma fonte de tensão contínua com uma gama de 6 a 20V. Este Arduino possui 14 entradas ou saídas digitais das quais 6 podem ser saídas PWM e 8 entradas analógicas de 10 bits o que permite uma resolução com 1024 valores. Tem memória flash de 32 KB memória SRAM de 2KB e EEPROM de 1KB. A frequência *clock* é de 16MHz.

No âmbito de este projeto foram usadas 4 entradas analógicas, uma para cada sensor de distância e 7 entradas digitais, 6 para o leitor LCD e uma para o botão de calibração.

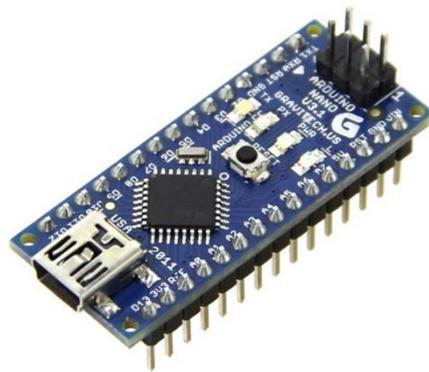


Figura 2-Arduino Nano AT328

## 4.3. LCD 16x2

---

O *display* utilizado neste projeto é um LCD com 16 colunas por 2 linhas com iluminação de fundo azul e escrita em branco, possui um controlador HD44780 e a interface com o Arduino é realizada basicamente por 4 pinos de dados e 2 de controlo.



Figura 3-LCD 16x2

## 5. Implementação da Hardware

---

Como a saída analógica dos sensores age como uma fonte de corrente e as entradas analógicas do Arduino nano fazem leitura de tensão, foi necessário converter a corrente das saídas dos sensores numa tensão proporcional a corrente, com uma gama de tensão de forma a aproveitar totalmente da resolução da entrada analógica. Para isso foi usado uma resistência em série com a saída analógica do sensor. O valor da resistência foi calculado dependentemente da corrente máxima do sensor (20mA) e da tensão máxima descegada, neste caso 5V.

$$R_s = \left( \frac{5V}{0,02A} \right) = 250\Omega \quad (1)$$

Como não é possível obter resistências de 250Ω foram usadas duas resistências, uma de 150 e uma de 100Ω com erro de 1% garantindo assim uma boa precisão na leitura.

Para reduzir frequências parasitas na tensão medida na resistência foi implementado um filtro passa-baixo com uma frequência de corte de 250Hz. O valor da resistência foi escolhido de forma a permitir o cálculo de um condensador com valor baixo, permitindo o uso de condensadores cerâmicos, mais indicados para este tipo de aplicação. O valor da resistência escolhida foi de 10kΩ e o condensador foi calculado através da seguinte fórmula.

$$C_F = \left( \frac{1}{f * 2\pi * R_F} \right) = \left( \frac{1}{250Hz * 2\pi * 10k\Omega} \right) = 63,6pF \quad (2)$$

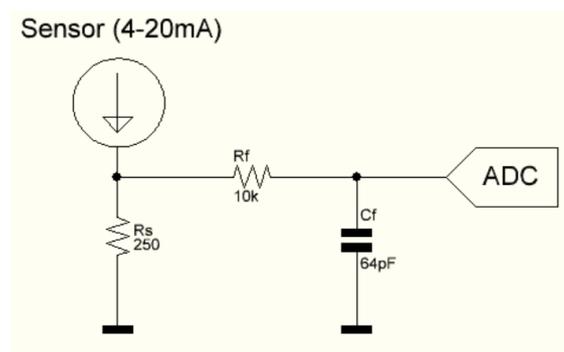


Figura 4-Resistência queda de tensão e filtro passa baixo

Após testes realizados em placa branca e os resultados das leituras serem satisfatórias, foi desenvolvido um PCB usando o *software eagle* para a realização do layout.

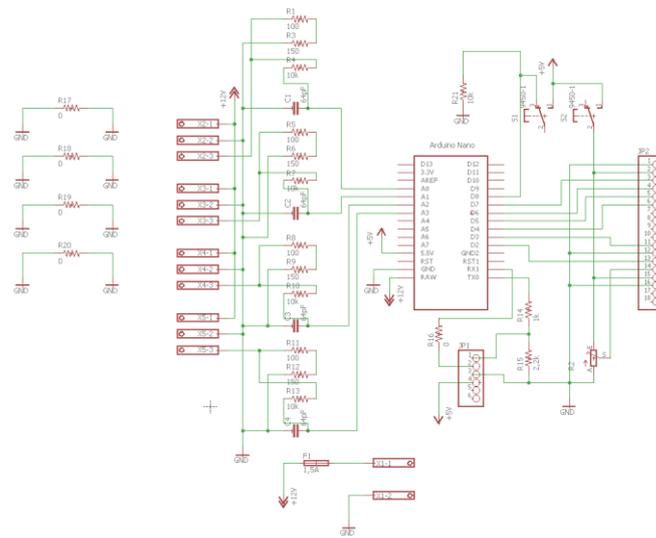


Figura 5- Esquema realizado em eagle

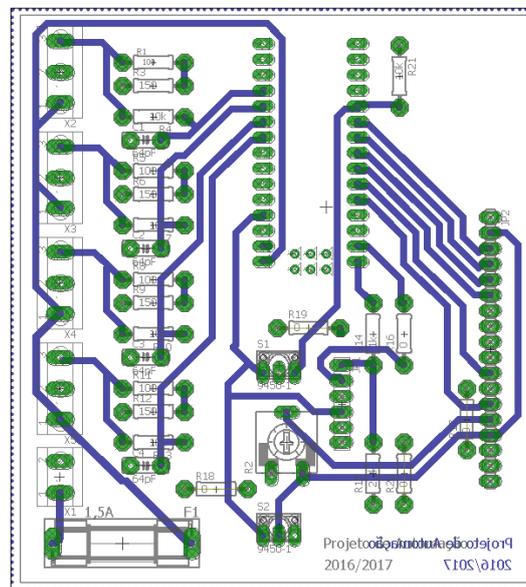


Figura 6- layout do PCB realizado em eagle

Este PCB engloba o tratamento de sinal dos sensores, o Arduino nano, um *socket* para a ligação de um LCD 16x2 com um potenciômetro R2 que possibilita o ajuste do seu contraste, um botão de calibração das leituras, um *socket* para a ligação de um módulo Bluetooth e um fusível de segurança F1 de 1,5A. A ligação dos sensores e da fonte de tensão é feita através de bornes de ligação por parafuso.

## 6. Aquisição de dados e conversão

Após realização do PCB, iniciou-se a aquisição de dados. Para isso, foram recolhidos 20 valores da queda de tensão na resistência de medição e da leitura da ADC do Arduino, para todos os sensores na gama de leitura de 200mm até 700mm.

	SensorA			SensorB			SensorC			SensorD			
32	distância sensor	tensão ADC	leitura ADC	dist calib	tensão ADC	leitura ADC	dist calib	tensão ADC	leitura ADC	dist calib	tensão ADC	leitura ADC	dist calib
33	200	1,014	202,37	200,02	1,018	202,35	200,22	1,012	202,35	200,20	1,02	202,35	200,08
34	225	1,216	242,42	224,60	1,217	242,42	224,88	1,219	243,4	225,39	1,22	242,42	224,78
35	250	1,421	283,48	249,80	1,416	282,5	249,54	1,416	282,5	249,38	1,42	283,48	250,08
36	275	1,624	324,54	275,00	1,62	323,56	274,81	1,618	323,56	274,58	1,62	323,56	274,78
37	300	1,826	365,59	300,20	1,825	364,61	300,07	1,822	364,61	299,77	1,82	364,61	300,08
38	325	2,029	406,65	325,40	2,025	405,67	325,34	2,026	405,67	324,96	2,03	405,67	325,39
39	350	2,229	446,73	350,00	2,226	445,75	350,01	2,227	446,73	350,16	2,23	445,75	350,09
40	375	2,435	488,76	375,80	2,43	486,8	375,27	2,433	488,76	375,95	2,43	485,83	374,79
41	400	2,633	527,86	399,80	2,631	526,88	399,93	2,632	527,86	399,94	2,63	526,88	400,09
42	425	2,839	568,92	425,00	2,831	566,96	424,60	2,832	568,92	425,13	2,83	566,96	424,79
43	450	3,037	608,99	449,60	3,035	608,02	449,87	3,035	608,99	449,72	3,03	608,02	450,10
44	475	3,239	650,05	474,80	3,239	650,07	475,75	3,235	650,05	474,91	3,23	648,10	474,80
45	500	3,443	691,11	500,00	3,435	689,15	499,79	3,437	691,11	500,11	3,44	689,15	500,10
46	525	3,643	731,18	524,60	3,636	729,23	524,46	3,638	731,18	524,69	3,64	729,23	524,80
47	550	3,845	772,24	549,80	3,843	771,26	550,33	3,84	772,24	549,89	3,84	770,29	550,10
48	575	4,04	813,3	575,00	4,05	811,34	574,99	4,04	813,3	575,08	4,05	811,34	575,40
49	600	4,24	854,35	600,20	4,25	851,42	599,66	4,23	853,37	599,67	4,25	851,42	600,10
50	625	4,44	894,43	624,80	4,45	892,48	624,92	4,44	894,43	624,86	4,45	891,50	624,80
51	650	4,65	935,49	650,00	4,66	933,53	650,19	4,64	935,49	650,06	4,65	932,55	650,10
52	675	4,85	976,54	675,20	4,86	973,61	674,85	4,84	976,54	675,25	4,86	972,63	674,80
53	700	5,05	1000	689,60	5,06	1000	691,09	5,04	1000	689,64	5,06	1000,00	691,67

Figura 7-Dados recolhidos e valores de leitura de distância convertida

A queda de tensão a entrada da ADC varia de 1 a 5V para distâncias de 200 a 700 mm. Esta variação corresponde a, aproximadamente, 200 a 1000 valores na ADC. Foi necessário determinar a equação da reta que relaciona a distância medida pelo sensor com os valores da ADC. Para isso, foram realizados os gráficos da figura 9 em *excel*. Os valores da tabela na figura 8 demonstram a linearidade da relação dos valores dos sensores em função da leitura da ADC.

<i>Estatística de regressão</i>	
R múltiplo	0,99999804
Quadrado de R	0,99999608
Quadrado de R ajustado	0,999995862
Erro-padrão	0,300855801
Observações	20

Figura 8-Valores de regressão para um dos quatro sensores

Na aquisição dos valores da ADC notou-se que em determinadas medições apareciam flutuações nos valores lidos da ADC sem haver variação da queda de tensão nas entradas analógicas. Este fenómeno é causado pela resolução da ADC do Arduino. As ADC's do Arduino têm uma resolução de 10 bits o que equivale a 1024 valores.

Inicialmente fez se uma aquisição de dados usando a gama total dos sensores, o que equivale a 1000mm. Numa segunda fase, diminuiu-se a gama de leitura para 500mm (de 200mm a 700mm) o que resultou numa melhor precisão na leitura e numa redução nas flutuações devido ao “efeito de escada” da ADC.

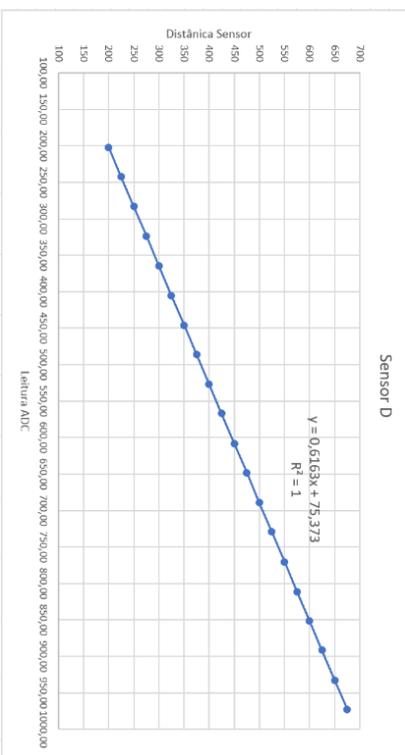
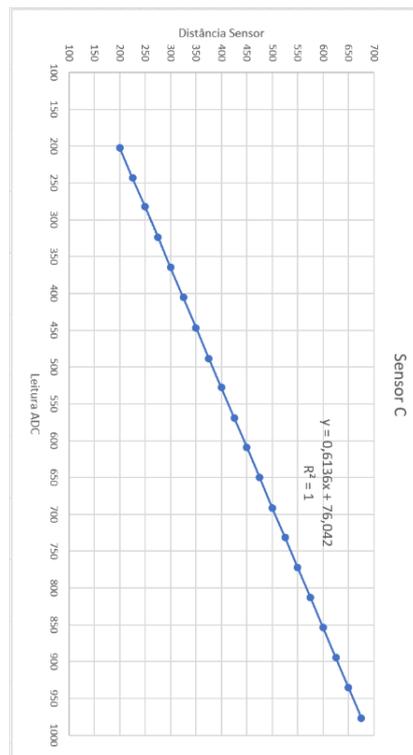
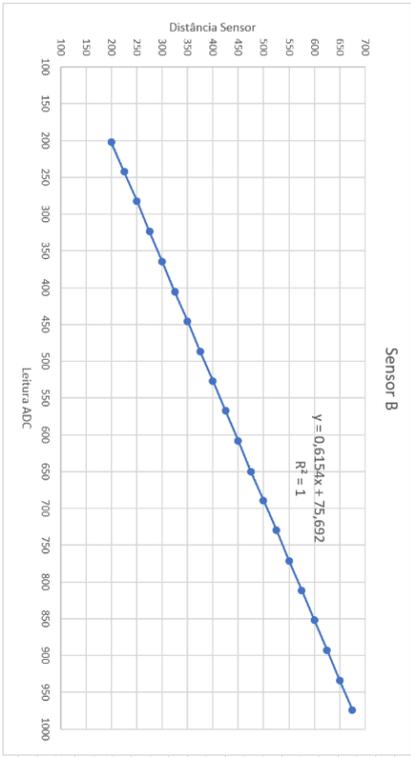
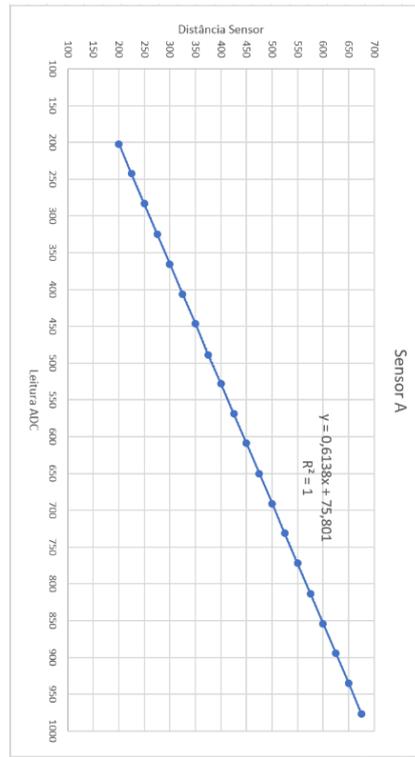


Figura 9-Gráficos da relação Distância Sensor com Leitura ADC

## 7. Implementação do programa no Arduino nano

---

Após a aquisição dos valores das ADC's foi necessário realizar um programa par obter os valores desejados neste projeto, que são os ângulos de inclinação *pitch*, no caso em que o caro esta sujeito a aceleração que seja a travagem ou ao arranque, e *roll*, quando o caro está sujeito a aceleração no caso de alteração da sua trajetória.

Para isso começou por se converter a leitura dos sensores com o auxílio das equações das retas determinadas no capítulo anterior. Agora que os valores das variáveis de distância correspondem à leitura dos sensores, realizou-se a calibração das leituras permitindo uma leitura a partir de um valor de referência, neste caso, o valor zero. Quando o botão de calibração é definido o *set-point* que irá ser subtraído a todos os valores lidos.

De seguida, graças à biblioteca *math.h*, realizaram-se os cálculos de trigonometria para a obtenção dos ângulos. O calculo realizado foi,

$$\hat{\text{Angulo de inclinação}} = \left( \frac{\text{soma da leitura dos sensores}}{\text{distância entre sensores}} \right) = \left( \frac{\text{lado oposto}}{\text{hipotenusa}} \right)$$

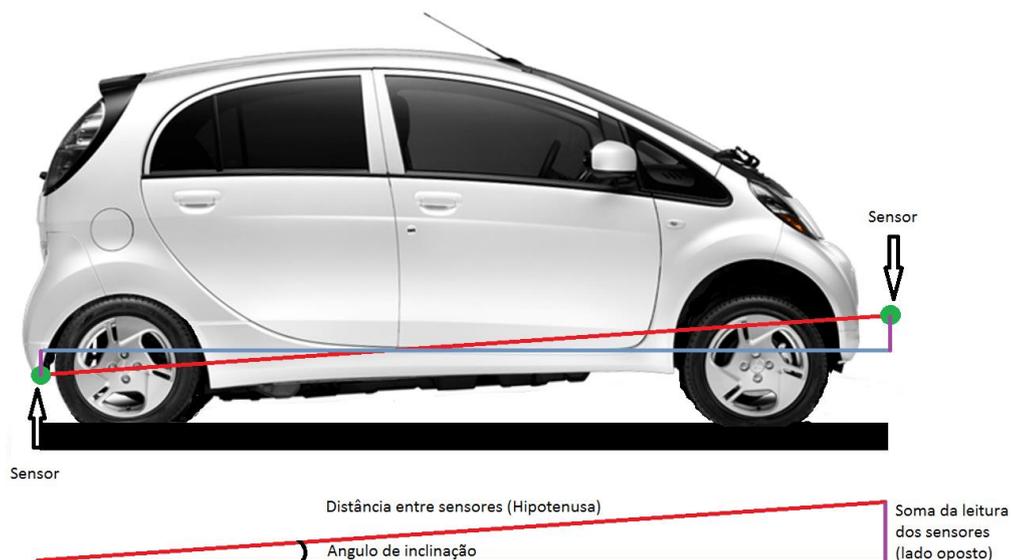


Figura 10-Ilustação da obtenção dos ângulos

Na figura 11 pode se ver como o calculo dos ângulos foi realizado através no programa Arduino,

```
//Inclinação para a frente*****  
if ((distance1_cut <= 0) && (distance3_cut >= 0))  
{  
  pitch_1 = (asin((-distance1_cut + distance3_cut) / dist_1_3))*57.2957;  
}  
//Inclinação para a trás  
else  
{  
  pitch_1 = (asin((-distance1_cut + distance3_cut) / dist_1_3))*57.2957;  
}  
  
//Inclinação para a frente*****  
if ((distance2_cut <= 0) && (distance4_cut >= 0))  
{  
  pitch_2 = (asin((-distance2_cut + distance4_cut) / dist_2_4))*57.2957;  
}  
//Inclinação para a trás  
else  
{  
  pitch_2 = (asin((-distance2_cut + distance4_cut) / dist_2_4))*57.2957;  
}  
  
  
//Inclinação para a direita*****  
if ((distance1_cut <= 0) && (distance2_cut >= 0))  
{  
  roll_1 = (asin((-distance1_cut + distance2_cut) / dist_1_2))*57.2957;  
}  
//Inclinação para a esquerda  
else  
{  
  roll_1 = (asin((-distance1_cut + distance2_cut) / dist_1_2))*57.2957;  
}  
  
//Inclinação para a direita*****  
if ((distance3_cut <= 0) && (distance4_cut >= 0))  
{  
  roll_2 = (asin((-distance3_cut + distance4_cut) / dist_3_4))*57.2957;  
}  
//Inclinação para a esquerda  
else  
{  
  roll_2 = (asin((-distance3_cut + distance4_cut) / dist_3_4))*57.2957;  
}
```

Figura 11-Programa para a obtenção dos ângulos

Distance1\_cut e Distance2\_cut correspondem à distância lida pelos sensores da frente e Distance3\_cut e Distance4\_cut correspondem à distância lida pelos sensores de trás. Dist\_x\_x corresponde a distância entre sensores e 57,2957 é o resultado da divisão de  $180/\pi$  na conversão de radianos para graus.

## 8. Montagem no veículo

---

Com a hardware e software realizada e o sistema testado em bancada, iniciou-se a montagem do sistema no carro. O primeiro passo foi fixar os sensores. Os sensores de trás foram aplicados aos suportes do para-choques traseiro como representado na figura seguinte,



Figura 12-Posição dos sensores traseiros

Os sensores da frente foram montados numa estrutura aplicada ao carro na realização de outro projeto como representado na figura seguinte,



Figura 13-Posição dos sensores da frente

De seguida foram aplicados os cabos que foram levados da mala, onde está colocada a central, até ao local de aplicação dos sensores. A hardware foi incorporado numa caixa em PVC onde foi montado o LCD em fachada e onde vão ligar os cabos dos sensores. A alimentação do sistema em 12V vem da caixa de alimentação situada na mala realizada noutro projeto.



Figura 14-Caixa PVC com hardware, ligações realizadas e em função

## 9. Testes de funcionalidade no veículo

---

O primeiro teste de funcionalidade consistiu em verificar se o sistema devolvia valores para os ângulos de forma correta. Usou-se, para isso, objetos com medida conhecida colocados debaixo dos sensores simulando inclinação do veículo. Os valores obtidos e aficados no LCD corresponderam aos valores calculados com calculadora. No entanto, deparou-se com um ligeiro problema de instabilidade dos valores, algo que já tinha sido reparado anteriormente, mas agora com maior frequência. Isto porque agora tanto os valores do *Pitch* como do *Roll* são dependentes dos quatro sensores, o erro de todos eles é acumulado.

Para determinar a importância das flutuações dos valores foi recolhido uma amostra de 500 valores das ADC's, das medidas convertidas e do *Pitch* e do *Roll*.

Os gráficos na figura 15 e 16 representam a frequência de valores da ADC e valores convertidos, entre o valor mínimo e valor máximo obtidos nas amostras para um dos sensores.

Como se pode ver não há acumulação de erro na conversão dos valores. A frequência de valores é a mesma em ambos os gráficos.

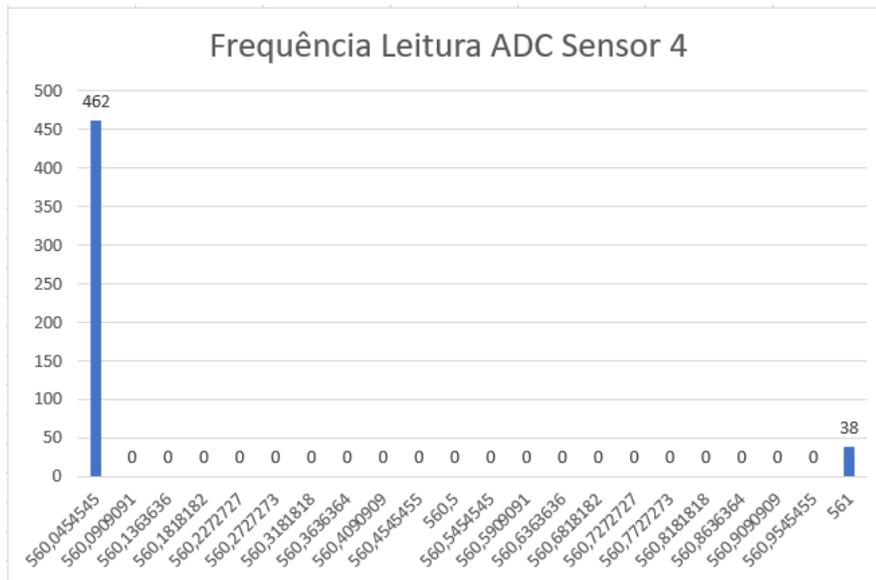


Figura 15-Frequência de ocorrência de valores ADC

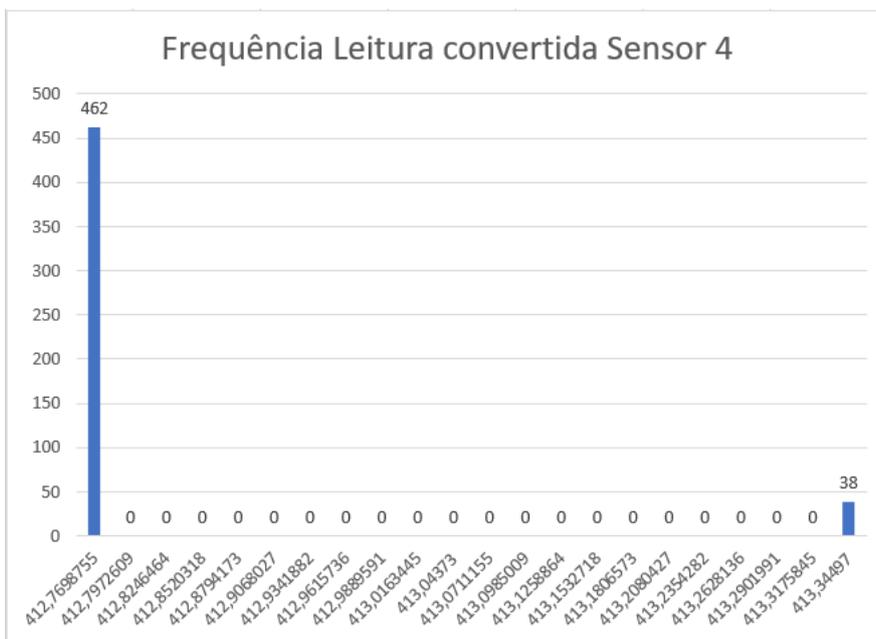


Figura 16-Frequência de ocorrência de valores ADC convertidos

Através dos gráficos nas figuras 17 e 18 pode-se ver a frequência dos valores do *pitch* e *roll* entre os valores mínimos e máximos da amostra. A saber que a gama de variação do erro é de 0,01533 graus para o *pitch* e 0,06869 graus para o *roll*.

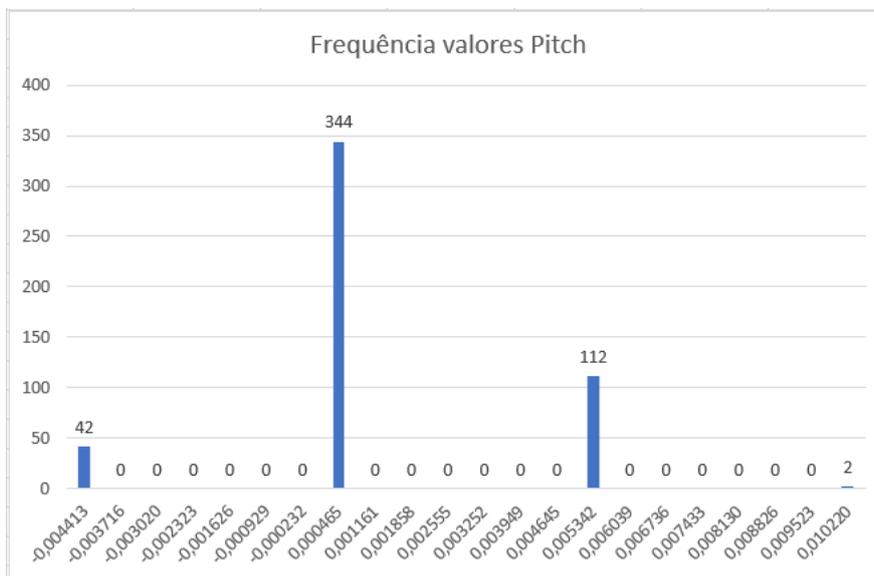


Figura 17-Frequência de ocorrência de valores *Pitch*

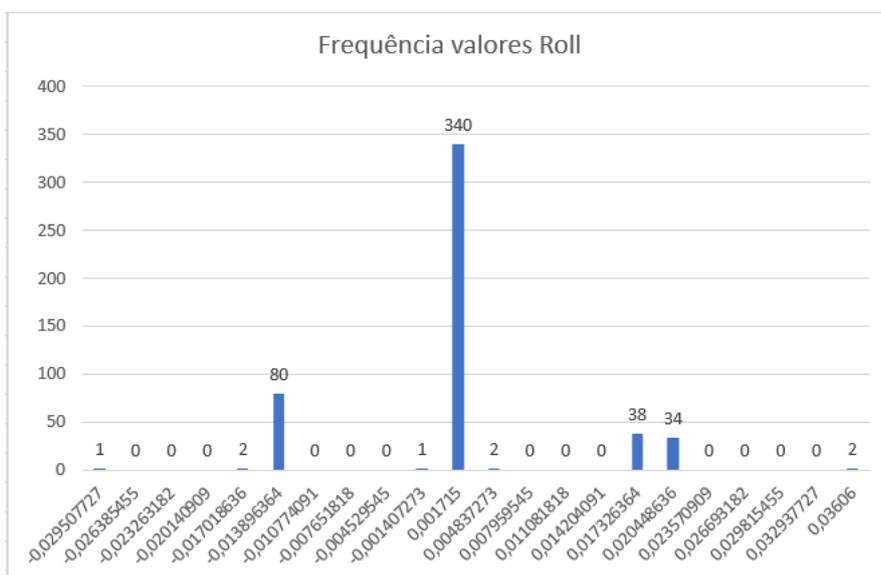


Figura 18-Frequência de ocorrência de valores *Roll*

Na tentativa de diminuir o erro do *pitch* e do *roll* foi feita, por software, uma média dos valores lidos. O resultado foi um aumento de valores de ocorrência, no entanto, a gama de erro diminuiu para 0,01148

graus no caso do *pitch* e 0,02465 graus no caso do *roll*. Os gráficos da frequência de ocorrência de valores com media de valores estão representados na figura 18 e 20.

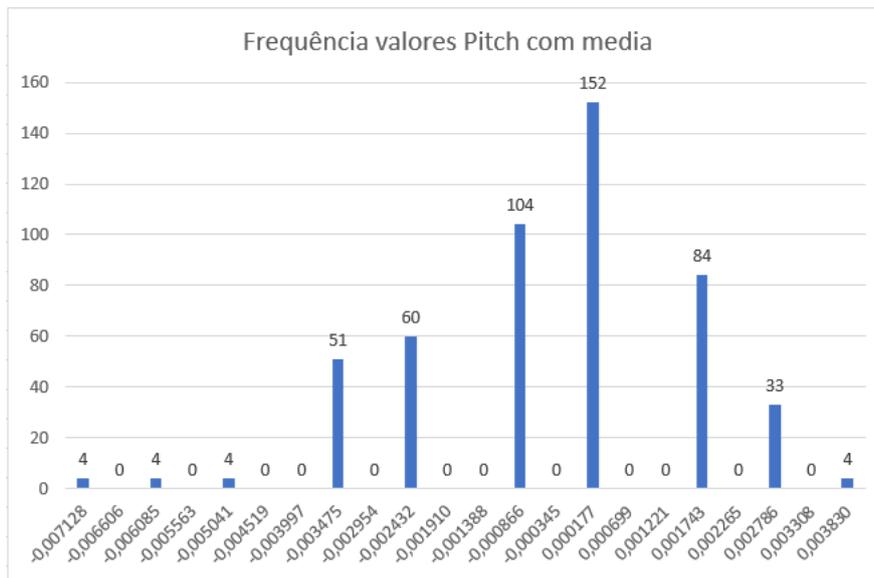


Figura 19-Frequência de ocorrência de valores *Pitch* com media de valores

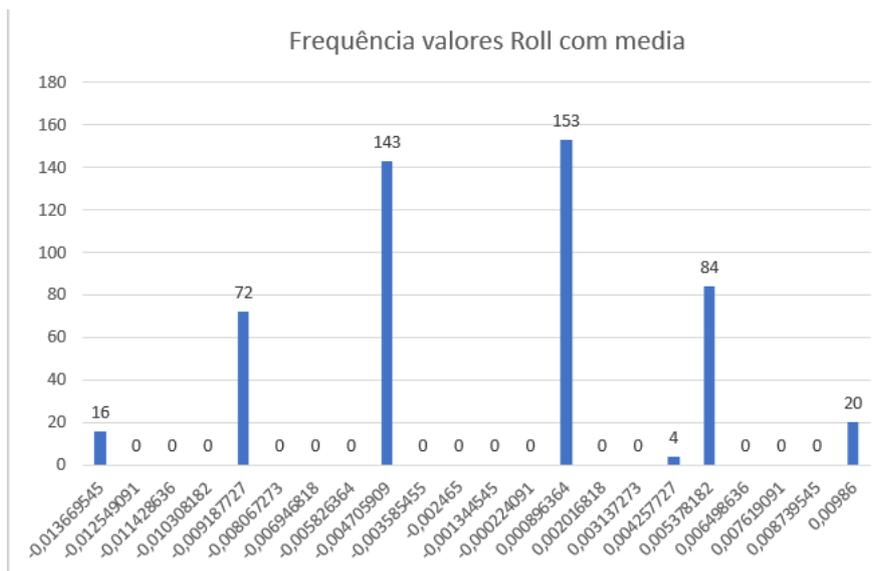


Figura 20-Frequência de ocorrência de valores *Roll* com media de valores

## 10. Comunicação por Bluetooth

Para além da comunicação através da porta usb, há também a possibilidade de comunicar com o Arduino através de um módulo Bluetooth montado no hardware. O módulo Bluetooth comunica com o Arduino através das portas série tx rx.

Foi realizado um teste de comunicação entre o PC e o Arduino usando para isso o terminal Tera Term, O Arduino foi programado de forma a permitir a visualização de valores constantemente e a efectuar a calibração através do envio da letra “C” a partir do terminal.

Na tentativa de realizar a comunicação com um *smartphone android* foi realizado uma aplicação usando a MIT appinventor. No entanto, até à data da escrita deste relatório não foi possível realizar a ligação Bluetooth entre o modulo e o *smartphone*.

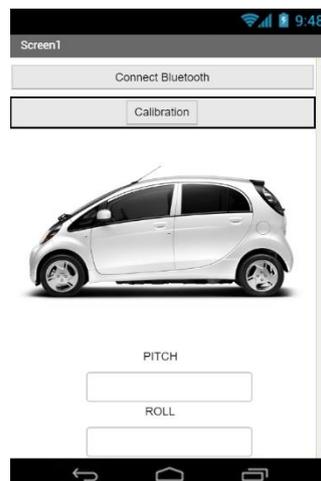


Figura 21- Aplicação android realizada na MIT appinventor

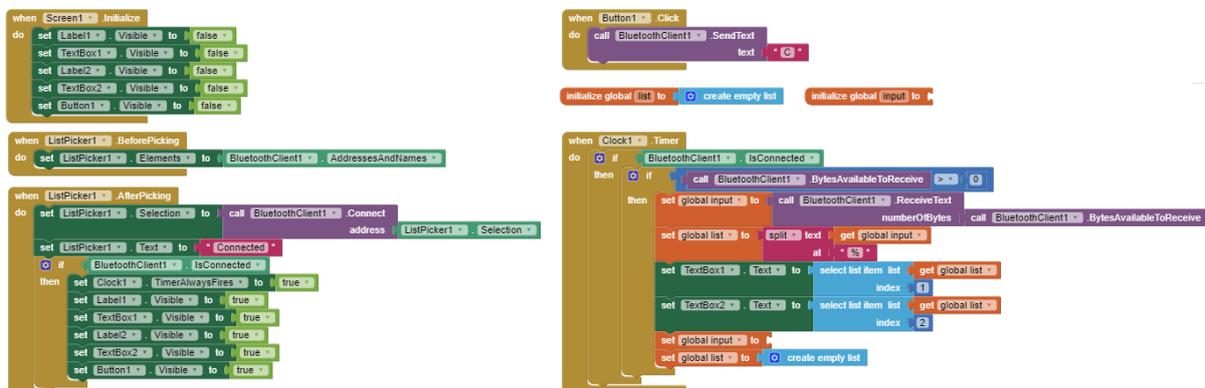


Figura 22-Diagrama de blocos do programa na MIT appinventor

## 11. Conclusões

---

Os objetivos iniciais foram todos realizados, no entanto, mais testes são necessários para uma melhor compreensão dos problemas existentes e a detecção de eventuais problemas ainda não detetados. Os testes realizados foram feitos com o carro em posição estática na oficina. O teste no exterior, com o carro em andamento, permitiria obter mais informação sobre o funcionamento do sistema e eventuais melhorias a realizar. Seria interessante a comparação do inclinómetro desenvolvido com um sistema equivalente do qual se conheça a fiabilidade. Será necessário avaliar se o erro induzido devido as ADC's e se é ou não relevante para o sistema no qual o inclinómetro irá operar.

Fica para concluir a comunicação Bluetooth a partir do smartphone que não foi conseguida com sucesso.