ADDETC – Área Departamental de Engenharia Eletrónica e Telecomunicações e de Computadores

LEIM -Licenciatura Engenharia informática e multimédia

Computação Física

Trabalho 3

Sensores e Atuadores

Turma:

LEIM23D

Trabalho realizado por:

Miguel Távora Nº45102

Luís Farinha Nº45147

João Cunha Nº45412

Docente:

Carlos Carvalho

Data de entrega:14/6/2019

Índice

Indice de Figuras

1.Introdução

Neste trabalho prático de Computação Física foi solicitada a realização do estudo de um sensor de giroscópio e de temperatura juntamente com um display 2x16.

Para que existisse comunicação simultanêa entre os dois dispositivos foram desenhados diagramas de atividades, com recurso a automatos não bloqueantes. O Arduino é o responsável pela comunicação com o sensor (configuração e leitura das medições) e com o display (configuração e afixação dos valores de velocidade ângular e temperatura calculados).

O sensor giroscópio tal como o nome indica devolve a velocidade angular em º/s. O sensor possui 3 eixos de ação sendo o x, y e z, do qual os valores variam conforme a rotação do sensor. Além de giroscópio o sensor possui ainda um sensor de temperatura pouco preciso que apenas indica a variação de temperatura ao qual foi sujeito.

A comunicação do arduino com o display LCD e com o sensor L3GD20 é feita recorrendo ao protocolo I^2C, dado que este protocolo serve para simplificar a comunicação entre muitos equipamentos. A sua montagem é relativamente simples dado que são necessários apenas dois cabos.

Também será exibida uma interface gráfica desenvolvida em Python. Para a construção da interface gráfica foi utilizada a biblioteca pygame. A partilha de dados é feita apartir de um canal série entre o Arduino e o programa python, utilizando a biblioteca Serial do python.

2.Desenvolvimento

2.1 Protocolo I^2C

O I^2C (Inter-Integrated Circuit Protocol) é um protocolo que funciona em barramento (bus), sendo preciso apenas dois fios por dispositivo para se conectarem ao protocolo, onde cada dispositivo terá o seu endereço único. O sensor giroscópio é identificado pelo endereço 0x69 e o LCD 2x16 pelo endereço 0x27.

A comunicação utilizada no bus é feita com base no modelo de "*master-slave*", onde pelo menos um dispositivo desempenha o papel de "*master*" e os restantes dispositivos o papel de "*slaves*". A função do "*master*" (Arduino) é coordenar a comunicação, sendo o responsável pelo envio de dados a um "*slave*" (LCD 2x16) ou consulta dados de outro *"slave"* (sensor giroscópio e temperatura). O I^2C é um protocolo que possui dois barramentos diferentes, o SDA (Serial Data, pino A4 no Arduino Uno) é um barramento bidirecional de comunicação, isto é, a linha que recebe e envia dados e outro barramento para controlar o envio e receção de dados, o SCL (Serial Clock, pino A5 no Arduino Uno) é o temporizador (Clock) que garante a fiabilidade da comunicação do SDA.

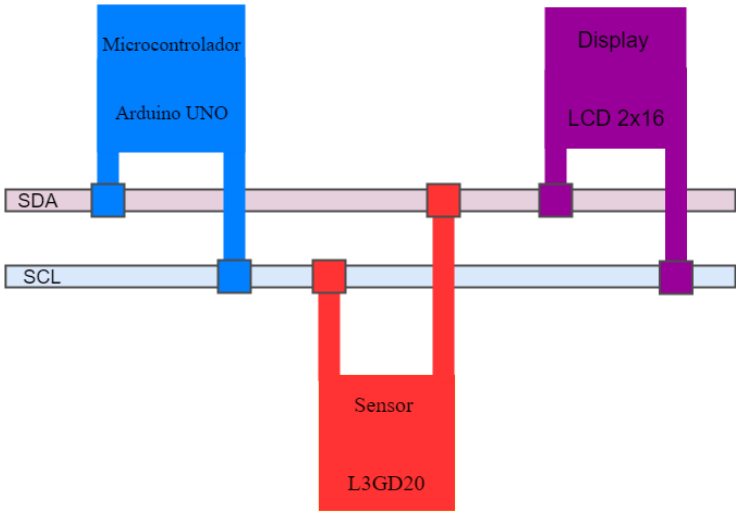

Figura 1 –Barramentos dos sinais digitais e dos dispositivos no I^2C

2.2.1 Diagrama de ligações do Arduino ao l3DG20 e Display

Conhecidos todos os integrantes do sistema em funcionamento, obtém-se o seguinte diagrama de ligações:

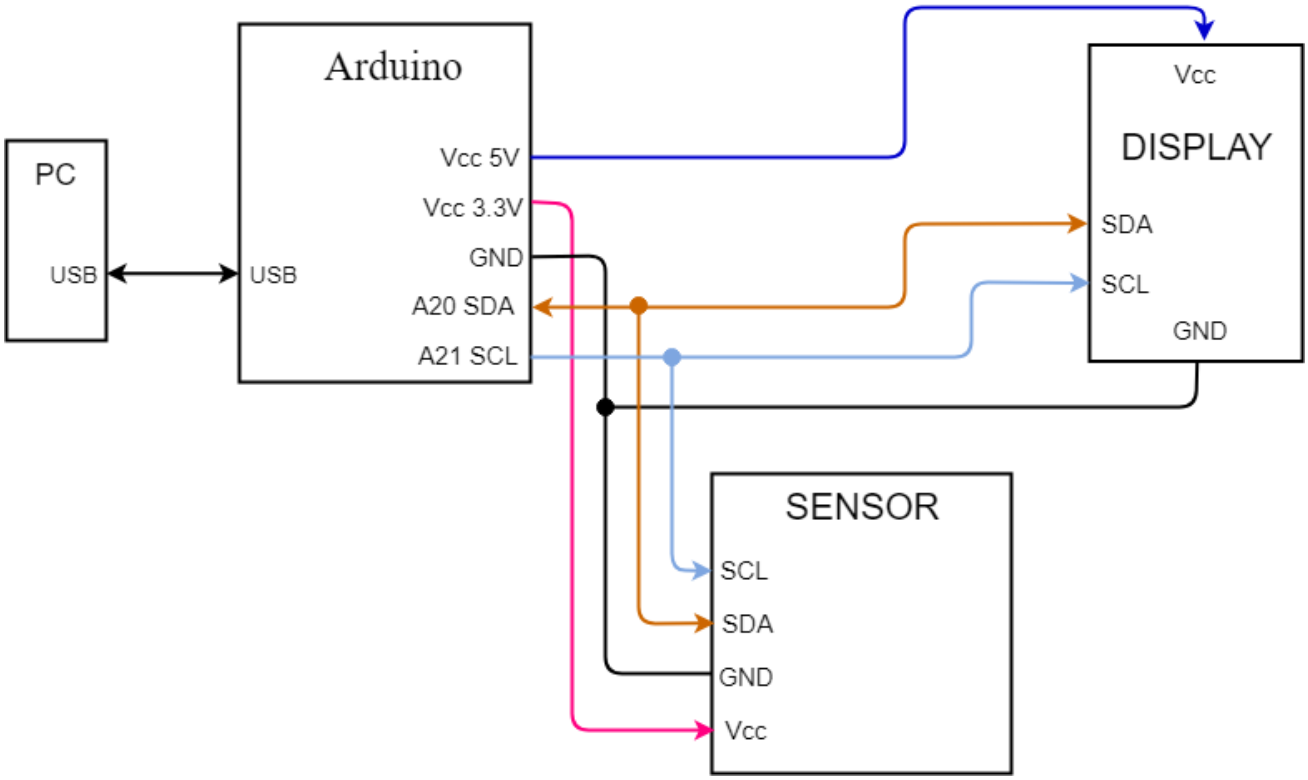

Figura 2 - Diagrama de ligações entre o Arduino e os sensores

2.2.2 Diagrama de atividades

Os diagramas de atividades permitem modelar as diferentes atividades do qual um sistema deve realizar para cumprir o seu propósito. Os diagramas permitem também representar as comunicações estabelecidas entre as tarefas. As vantagens da linguagem gráfica são as independências da plataforma da tecnologia e da linguagem de programação.

Pretende-se modelar um sistema que realize ciclicamente a leitura dos valores da velocidade ângular e da temperatura com um tempo de espera desprezável. Para isso o microcontrolador lê os valores do sensor e converte esses valores para os valores pretendidos e afixa os valores no display LCD2x16.

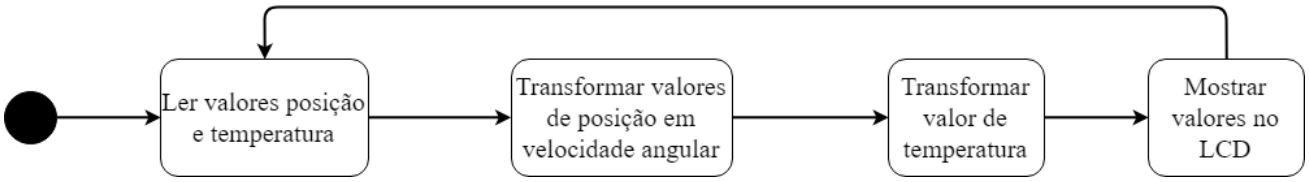

Figura 3 - Diagrama de atividades

A partir da observação do diagrama de atividades é possível verificar o funcionamento do sistema como um todo. Desta forma podemos visualizar as diferentes tarefas que são executadas e as condições que mantém o funcionamento constante da atividade. Através da atividade é possível a afixação dos valores de temperatura e velocidade angular dos eixos x, y e z.

Contudo, para as tarefas seres executadas ordenadamente é necessária uma troca de objetos. O que permite a uma atividade dar a vez a outra é a modificação do estado dos objetos trocados entre as atividades.

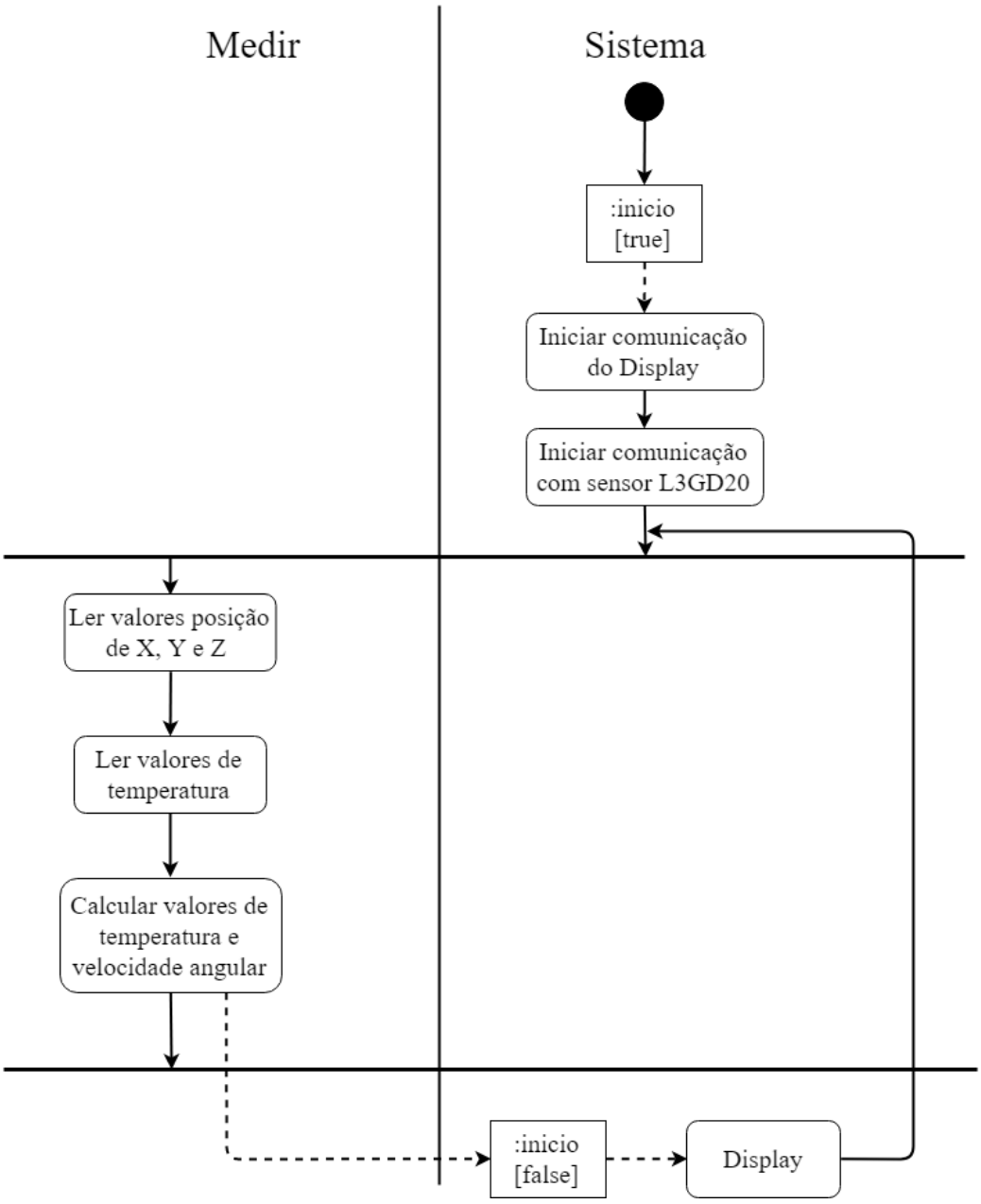

Figura 4 - Fluxo de objetos

Observando o fluxo de objetos é possível verificar a existência de um objeto (variável utilizada no código) chamado inicio. Quando o objeto possui o valor "*true"* ele inicia a comunicação com o display LCD e com o sensor. De seguida, une as duas comunicações, lê os valores de temperatura e da posição dos eixos. Quando os valores forem lidos calcula-se os respetivos valores de velocidade angular dos eixos de x, y e z e da variação da temperatura. Para não estar sempre a fazer os setup's a variável início toma o valor de false e são exibidas os valores no LCD. Após afixados os valores ele volta a ler e a calcular os valores do sensor e o processo é repetido ciclicamente.

2.3 Autómatos Não Bloqueantes

A implementação do diagrama de atividades anterior é possível com recurso a autómatos não bloqueantes, sem estado final. As características destes autómatos é o facto de o tempo de execução ser finito, mínimo e limitado à execução das ações de uma atividade. O facto de não terem estado final indica que todas as atividades estão ligadas entre si, através de condições de transição.

O processo de medição e afixação de resultados será executado recorrendo a dois autómatos: o autómato sistema e o autómato medir. O autómato sistema é responsável por tarefas como iniciar e exibir os valores no LCD. O autómato medir é responsável por realizar as leituras e os cálculos dos valores.

A comunicação entre autómatos é feita por intermédio da variável inicio, como referido anteriormente quando o seu valor é "*true*" inicia a comunicação com o LCD e com o sensor. Após feita a comunicação a variável inicio passa a ser *false* e não é realizada novamente a iniciação da comunicação.

2.3.1 Autómato Sistema

O autómato Sistema é responsável por iniciar as comunicações com os dois dispositivos e também pela exibição dos valores no display. Esta sequência de ações inicia a comunicação primeiramente com o display e de seguida com o sensor. Após estabelecida a comunicação o autómato aguarda a execução do autómato medir de leitura dos valores. Quando os cálculos são concluídos a variável início passa o seu valor a "*false*" e esse valor não é alterado mais nenhuma vez durante o código. Paralelamente os valores lidos do sensor são exibidos no LCD.

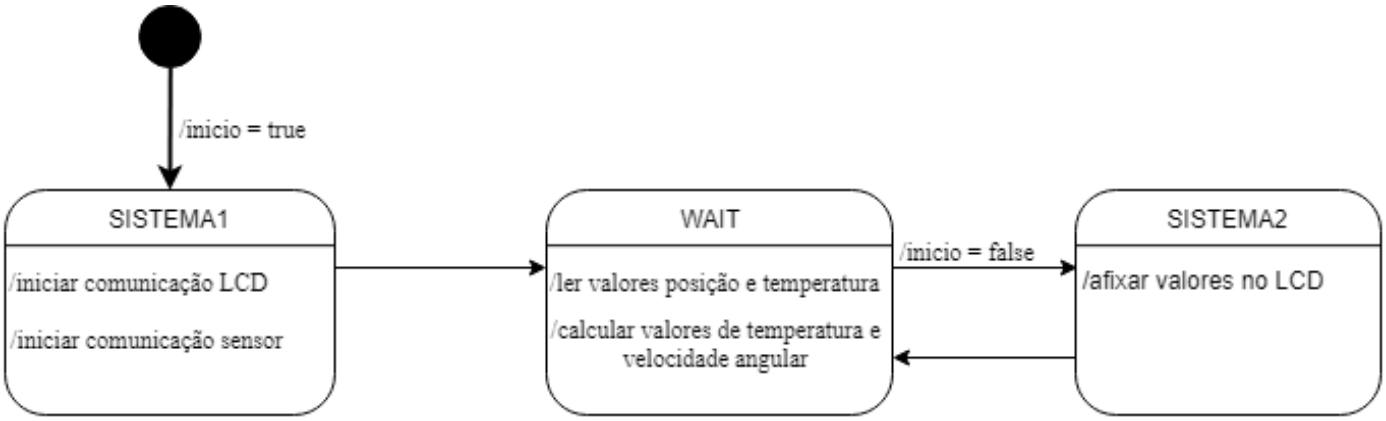

Figura 5 - Autómato Sistema

2.3.2 Autómato Medir

O autómato medir é o responsável por efetuar as leituras da temperatura e da velocidade angular. A sua execução inicia no estado de espera (WAIT) e permanece nesse estado até haver uma condição de transição, no caso particular, enquanto a comunicação não estiver estabelecida ou enquanto os valores não forem exibidos no LCD. Quando transita de estado efetua primeiramente as leituras das variações das frequências dentro do sensor e a leitura da temperatura e de seguida são calculados os valores exatos da temperatura e da velocidade angular dos 3 eixos. Aquando do retorno dos valores o autómato transita para o estado de espera até os valores serem afixados no LCD, quando afixados ele volta a ler e calcular esses mesmos valores e assim sucessivamente.

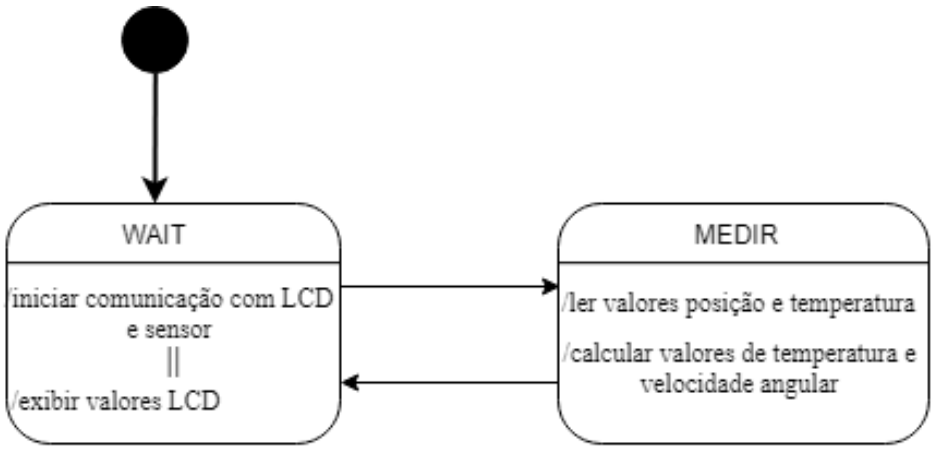

Figura 6 - Autómato Medir

2.4 Sensor L3GD20

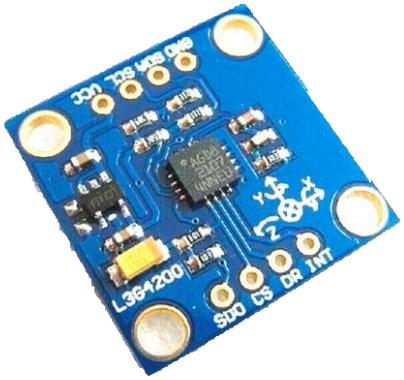

Figura 7 - Sensor L3GD20

O sensor L3GD20 contém além de um giroscópio digital um acelerómetro e uma bússola. Cada sensor fornece informação sobre três eixos no espaço, sendo os eixos de x, y e z totalizando assim 9-DOF(*Degrees of freedom*). Apesar de o sensor ser possuidor destes três sensores só será utilizado o giroscópio. Também é possível obter uma aproximação da temperatura utilizando o sensor, este tem a capacidade de detetar a variação da mesma, mas sem precisão. O valor que devolve, tem o significado de -1ºC/bit, pelo que é necessário utilizar uma fórmula matemática para obter a variação correta da temperatura.

2.4.1 Algoritmo para obter o ângulo de rotação do veículo com o L3GD20

O sensor trata-se de um dispositivo mecânico inercial sensível à rotação. Para a obtenção da velocidade angular o dispositivo é mantido a vibrar a uma frequência conhecida, quando essa vibração é perturbada estima-se a velocidade angular, a informação é feita segundo o contrário do sentido horário. A sensibilidade varia de acordo com os fins de escala do qual pode variar $8.75\text{x}10^{-3}$(º/s)/bit, $17.5\text{x}10^{-3}$ e $70\text{x}10^{-3}$ para as escalas 250º/s, 500º/s e 2000º/s, respetivamente.

Caso a velocidade angular seja constante é possível de obter o deslocamento, tendo um intervalo de tempo e segundo um eixo por:

$$\varphi_Z = \int_{t_0}^{t_1} \Omega_Z \, dt + \varphi_{Z0} = \Omega_Z.(t_1 - t_0) + \varphi_{Z0}$$

2.4.2 Método utilizado para cálculo do ângulo

Utilizando a velocidade angular é possível calcular por estimativa a amplitude do ângulo girado. No nosso trabalho, o Arduíno envia os valores da velocidade ângular para o serial de 0,04s em 0,04s(t).

$$\hat{A}ngulo = \hat{A}ngAnterior + VelociadeÂngular * (0.04)$$

Ou seja, a cada 0,04s é somado o ângulo

2.4.4 Métodos implementados (sensor L3GD20)

Este método configura o sensor de modo descrito na sua folha [3] (foi utilizada a escala de 250 dps):

- setupL3G420()

Após o sensor ter sido configurado, a leitura dos valores de X, Y, Z e Temperatura é feita pelos endereços específicos [3].

- getGyroValues()

Por último, após a leitura dos valores de X, Y, Z, e temperatura, estes são transformados como explicado na introdução:

- gyroCalc()

2.5 LCD 2x16

2.5.1 Explicação do funcionamento do display I2C

O LCD utilizado é 2x16, ou seja, 2 linhas por 16 colunas. Este tipo de LCD é bastante utilizado em equipamentos comerciais e industriais. A principal característica destes LCD's são os baixos requisitos visuais quando comparados a equipamentos como monitores. A interligação a microcontroladores é bastante simples, onde cada byte enviado pelo I^2C ele utiliza os pinos de D7 a D0. Os pinos D7 a D4 são para envio de dados, o D3 para retroiluminação (*backlight*) e os pinos D2 a D0 para controlo (E, RW e RS respetivamente).

Através das especificações dadas pelo fabricante do display LCD primeiramente é feita a configuração para funcionar a 4 bits, de seguida as 2 linhas por 16 colunas, display, cursor e *blink* ativos e posicionadas no cursor do display.

Figura 8 - Display LCD 2x16 e conversor

2.5.3 Métodos implementados (display)

Este método configura o display de modo descrito na sua folha [4]:

- DisplaySetup()

Este método apaga toda a informação no display 2x16 [4] (código: 0x01):

- DisplayClear()

Este método seleciona a primeira fileira do display ou na segunda para de seguida ser escrita informação de modo mais organizado [4]:

- DisplaySetCursor()

Método responsável por enviar 8bits de informação para o display;

- escreverDados8() = DisplayChar()

Permite imprimir no display 2x16 strings, que transforma uma string num array de “char” e envia carater a carater utilizando a função anterior:

- DisplayString()

Permite imprimir no display 2x16 números inteiros, transforma o número numa string e envia para a função anterior:

- DisplayInt()

2.7 Interface gráfica em Python

Ao iniciar o programa, o seguinte menu é aberto, permitindo ao utilizador começar ou sair.

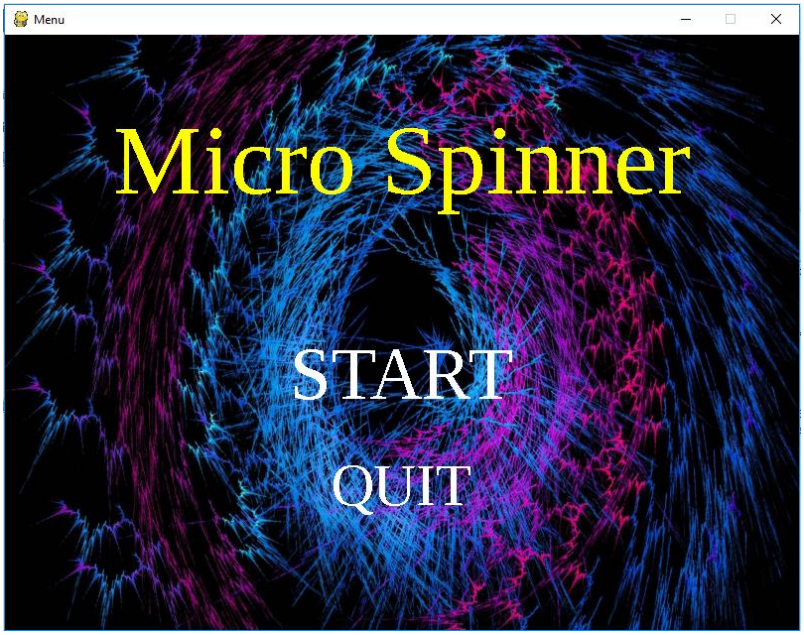

Figura 9 - TP3.py, menu

Em primeiro, foi criado uma parametrização que faz o carro andar a volta da pista, utilizando o ângulo Z do sensor, o carro anda mais rápido ou mais lento.

Figura 10 - TP3.py, interface 1

Em segundo, temos um carro que é controlado pela velocidade angular X e Y do sensor. Isto permite ao utilizador, "conduzir" o carro girando o sensor nos eixos do X e do Y. Para a rotação do carro (imagem), foi utilizado o angulo Z, conforme o ângulo do sensor relativamente ao eixo Z, a imagem do carro gira. Como podemos ver na figura, o ângulo Z está a 90º, oque provoca uma rotação de 90 graus na imagem do carro.

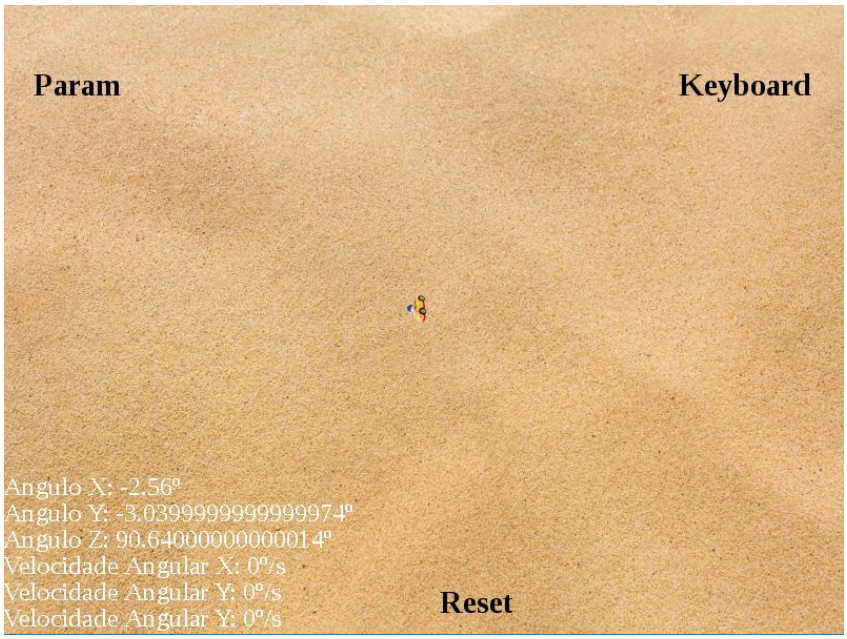

Figura 11 - TP3.py, interface 2

Por último, o programa tem um termómetro interativo que é atualizado de 15 em 15 segundos (enunciado). O carro é controlado pelo teclado oferecendo uma experiência diferente ao utilizador.

Figura 12 - TP3.py, interface 3

2.7.1 Comunicação série e sincronização na transferência de informação PC – Arduino

Quando o Arduino é ligado ao computador por meio de cabo USB (Universal Serial Bus) é estabelecido um canal de comunicação série onde são carregados os programas sendo assim possível a comunicação entre o Arduino e outro dispositivo através do Serial Monitor.

Quando o programa fui enviado para o Arduino, este imprime no Serial os valores da velocidade angular de X, Y, Z e temperatura.

Utilizando a biblioteca Serial no Python, é possível ler os valores enviados do Arduino em “tempo real”. Para o efeito foi utilizado a função readArduino() que faz a ligação do Python ao Serial do Arduino.

As principais linhas de código utilizadas são as seguintes:

- s = readArduino()
- if(s.inWaiting()>0):
- stringReceive(s)

2.9 Montagem

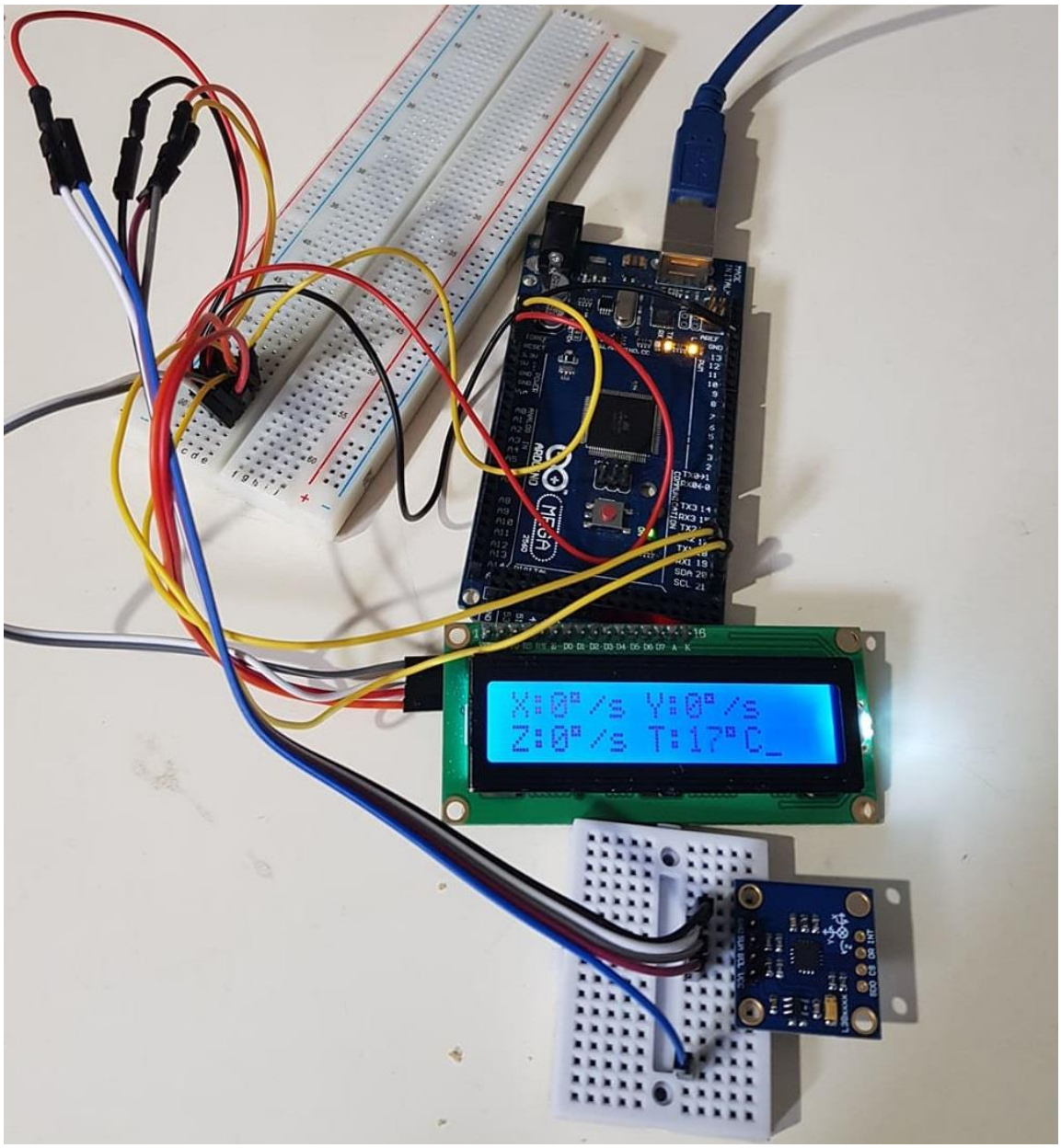

Figura 13 - Fotografia da montagem

3. Conclusão

Em suma, durante o presente o presente trabalho prático o grupo foi capaz de adquirir as seguintes competências:

- Compreensão e utilização do protocolo de comunicação I^2C.
- Utilização do sensor L3GD20 para leitura de valores de velocidade angular e temperatura.
- Construção do diagrama de atividades para verificar a sequência das diferentes tarefas.
- Dimensionamento dos autómatos necessários para executar tarefas de medição e afixação de valores no display de forma pseudo-paralelas ao utilizador, tendo variáveis booleanas para intermédio da comunicação.
- Desenvolvimento de um Serial Monitor em Python, no qual foi feita uma interface gráfica com o auxílio da biblioteca Serial para comunicar entre o computador e o Arduino.

Tal como foi referido anteriormente, o trabalho tinha como principal objetivo o máximo de fluidez na parte da interface gráfica, pelo que o sistema não tem nenhum tempo de espera suficientemente grande para causar impacto nos valores lidos do sensor. Sendo que o utilizador terá a precessão de que os valores obtidos são instantâneos.

4. Bibliografia

- [1] Folhas de Computação Física, Jorge Pais, 2018/2019
- [2] Folhas de Computação Física, Carlos Carvalho, 2019
- [3] Data sheet L3GD20, Digital three-axis digital output gyroscope (2013)
- [4] Data sheet display LCM1602, LCD MODULE SPECIFICATION

Anexos/Código

***Separar código TP3

***Código LCD

***Código Sensor

***Código Python

Implementação do automato correspondente ao diagrama de atividades/estados
Implementação dos métodos do Display I2C
Implementação da Interface gráfica em Python
Comunicação e sincronização entre o Arduino e o Computador