Relatório Final de Pesquisa de Iniciação Científica

Título do Projeto:

Estudo do Arduino Yún para o Controle dos Propulsores de uma Embarcação não Tripulada

Identificação

UERJ
Faculdade de Engenharia — FEN
Departamento de Eletrônica e Telecomunicações — DETEL

Período deste Relatório: agosto de 2017 a julho de 2018

Equipe:

Bolsista PIBIC/UERJ: Felipe Gomes S. Souza — Matrícula: 2013.20514111

Professor Orientador: José Paulo Vilela Soares da Cunha — Matrícula: 32.640-5

Local de desenvolvimento do projeto: Laboratório de Controle e Automação da Faculdade de Engenharia Elétrica

Local e data: Rio de Janeiro, 30 de agosto de 2018

Resumo

Este relatório evidencia o processo de criação e configuração de uma página HTML (Hyper Text Markup Language- Linguagem de marcação de Hipertexto), que exibe dados atualizados de alguns sensores instalados numa embarcação não tripulada, utilizando o Arduino Yún como servidor.

1 Introdução

Este projeto surgiu da necessidade desafiadora que é estudar nossos ecossistemas em especial os mares, rios e a atmosfera, devido a grandiosidade do nosso planeta. Alguns ambientes são difíceis de serem acessados ou podem representar danos á saúde dos seres humanos caso haja um contato prolongado, sendo assim o estudo de embarcações não tripuladas torna-se importante para que a exploração seja realizada (SOUZA,2016).

No trabalho (CRUZ, 2017) surge a ideia da utilização do Arduino Yún, uma vez que ele é o composto por um módulo de comunicação com redes *Ethernet* e *Wi-Fi* (Wireless Fidelity - Fidelidade sem fio), um microcontrolador com sistema operacional Linux e um microcontrolador para interface com atuadores e sensores.

2 Objetivo

O objetivo desse trabalho é criar uma página HTML que monitore as variáveis de estado dos propulsores de uma embarcação não tripulada. A implementação de sensores será realizada, uma vez que possibilita a criação de um sistema de monitoramento.

3 Considerações Iniciais

O Arduino Yún possibilita que o usuário utilize-o como servidor. Quando conectado via *Wi-Fi*, o Arduino Yún permite a criação de uma página HTML, que será carregada em um *SD card*, sendo possível acessá-la por meio de um navegador. Na página HTML as informações dos sensores são exibidas em navegadores de Internet e atualizadas com o auxílio de uma biblioteca minimalista do JavaScript , o zepto.min.js.

Em um primeiro instante, nos dedicamos a criação de um algoritmo no Arduino Yún, que foi iniciado no trabalho (CRUZ,2017) e que está disponibilizada no Apêndice A, para exibir as variáveis de estados dos propulsores. Foram implementados alguns sensores, conforme realizado no trabalho (ROSARIO, 2013), que além de fundamental para o sistema de monitoramento, serão futuramente importantes ao realizar o sistema de controle e de proteção.

Um *SD card* foi configurado no Arduino Yún para realizar corretamente a leitura dos dados pelo algoritmo. No Apêndice B encontra-se o algoritmo da criação da página HTML que também foi carregada no *SD card*. Desta maneira é possível realizar a comunicação

a distância com o Arduino e obter informações sobre os sensores utilizados. Na Figura 1 encontra-se um esquemático do trabalho realizado.

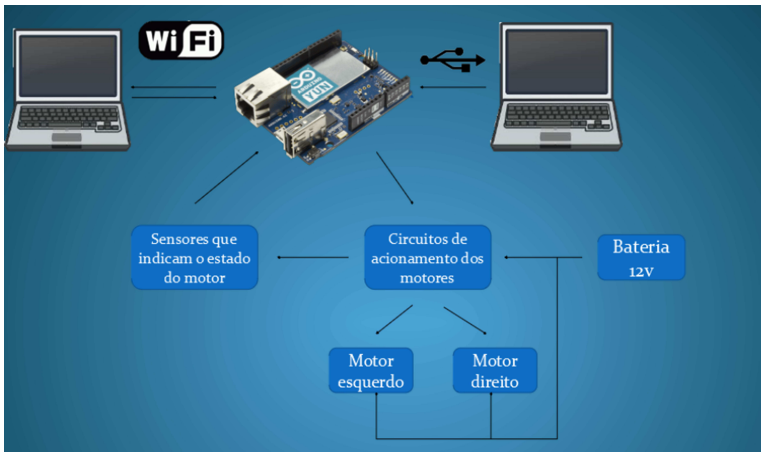

Figura 1: Esquemático do Projeto, adaptado de (CRUZ,2017)

4 Configuração do Arduino Yún

4.1 Configuração do Arduino

Na etapa inicial devemos realizar a configuração do Arduino Yún acessando: http://192.168.240.1 ou `http://arduino.local.com`. Será aberta uma página, conforme mostra a Figura 2 , solicitando um *password*, que é "**arduino**".

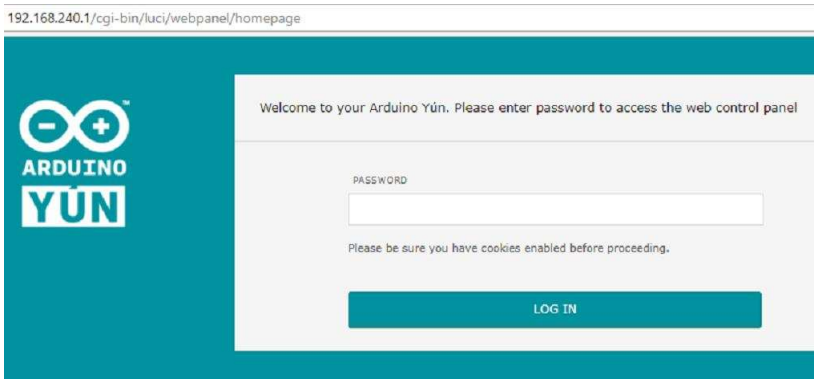

Figura 2: Página inicial de configuração do Arduino Yún

Ao entrar na página haverá sobre o Arduino, uma parte será destinada a configuração do mesmo, conforme mostra a figura 3. Terminada a configuração clique em "**Configure**".

Realizada a configuração do Arduino e terminados os algoritmos presentes no Apêndice A e Apêndice B, devemos realizar a configuração do *SD card*.

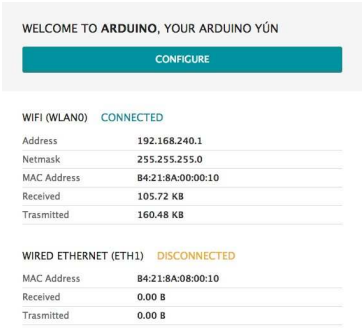

Figura 3: Página da configuração do Arduino Yún

4.2 Configuração do *SD card*

O*SD card* deve conter inicialmente uma pasta denominada"**Arduino**", que deverá conter uma pasta vazia denominada "**meualgoritmo**", que deve ter o mesmo nome dado ao algoritmo criado no Arduino. Como nomeamos o nosso algoritmo de "**html2**", deste modo, a pasta vazia contida na pasta "**Arduino**"será denominada de "**html2**".

A pasta estará no caminho: SD/arduino/www/meualgoritmo. Uma vez que se o Arduino estiver conectado ao *Wi-Fi*, todas as informações presentes na pasta do algoritmo serão transferidas automaticamente para o *SD card*, ao efetuar o carregamento do algoritmo. A Figura 4 mostra a estrutura inicial da página do *SD card*.

Figura 4: Estrutura da página do *SD card*

Assim que o algoritmo for carregado via *Wi-Fi* para o Arduino, a estrutura presente na Figura 4 será automaticamente copiada para o *SD card*. Desta maneira o *SD card* terá a mesma estrutura da página do *Sketch*.

4.3 Configuração da pasta do Arduino

Se o algoritmo já estiver salvo, a pasta do Arduino pode ser acessada em **Sketch** > **Mostrar a página do Sketch**, conforme a Figura 5.

Figura 5: Pasta do Arduino Yún

O Arduino Yún pode ser conectado tanto via USB quanto via *Wi-Fi*, no entanto, para que as informações sejam gravadas no *SD card*, a conexão deve ser efetuada via *Wi-Fi*. Para realizar a conexão via *Wi-Fi*, o Arduino deve estar conectado a porta de rede, conforme mostrado na figura 5. Caso seja o primeiro carregamento, um será requerido e a chave de acesso é "**arduino**".

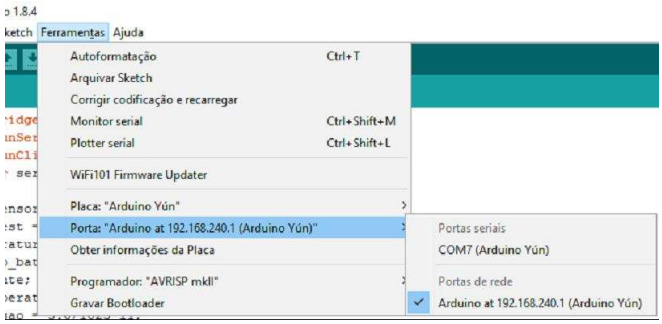

Figura 6: Conectando o Arduino Yún na porta *Wi-Fi*

A pasta do Algoritmo deverá apresentar a estrutura presente na Figura 6. O Zepto, que é a biblioteca minimalista do JavaScript, deverá estar contido na pasta"**www**". O Zepto pode ser baixado diretamente do site http://http://zeptojs.com ou copiado do exemplo "**TemperatureWebPanel**". O nome dado à página HTML foi "**Index.Html**".

Figura 7: Estrutura da pasta do Arduino

4.4 Acesso à página HTML

A página HTML pode ser acessada em `http://192.168.0.193/sd` e será possível ver os diretórios do *SD card*. Clicando no diretório "**html2**"temos acesso à página HTML criada, conforme mostrado na as Figura 7 e 8

Figura 8: Figura da Esquerda

Figura 9: Figura da Direita

5 Sensores

Os sensores representam uma parte fundamental do projeto, pois são dispositivos que ao receberem um estímulo do ambiente respondem com um sinal elétrico (SOUZA, 2016). Três tipos de sensores foram usados neste projeto: sensor de temperatura, sensor de tensão e sensor de corrente. Em um âmbito geral, os propulsores da embarcação são motores de corrente contínua que possuem algumas características dependentes da corrente e da tensão aplicada, que são o torque e a velocidade, respectivamente.

Desta maneira, ao realizar o controle dos propulsores da embarcação não tripulada, deverá ser levada em conta as informações provenientes do sensor de corrente e do sensor de tensão. O sistema de proteção dos motores, que serão implementados futuramente, utilizam os sensores para detectar possíveis falhas, por exemplo, caso haja travamento dos propulsores certamente um sensor de temperatura irá notar um sobreaquecimento, caso haja um curto-circuito certamente o sensor de corrente irá notar uma corrente elevada e etc.

5.1 Sensor de temperatura

Neste projeto o sensor analógico de temperatura utilizado foi o LM35, conforme utilizado na trabalho de (SOUZA,2016). Este sensor opera numa faixa de temperatura de -55 °C a 150 °C, com precisão de aproximadamente 0,250 °C em uma temperatura ambiente. Cada $10mV$ medidos correspondem a aproximadamente 10 °C,portanto, o LM35 opera com uma

relação de 10mV/0°C. Nas referências encontra-se link do manual do LM35, caso haja necessidade de mais informações.

Figura 10: Sensor de Temperatura LM35, extraído do manual do LM35

Para obter a temperatura em graus Celsius, o sensor lê um valor, que pode variar de 0 até 1024, divide esse valor por 1024 e multiplica o resultado por 5. Como um graus Celsius corresponde 10mV, portanto basta multiplicar o valor do sensor por 100. Sendo assim, podemos elaborar a seguinte equação:

$$T = \frac{V_{REF} \cdot E_{A5}}{2^{10} \cdot 0,01} \tag{1}$$

Onde,

T é a temperatura medida em graus Celsius;
V_{REF} é o valor de Referência;
E_{A5} é o leitura analógica foi na porta 5 do Arduino Yún.

5.2 Sensor de corrente

O sensor utilizado foi o ACS756SCA-100B-PFF-T. Esse sensor é capaz de medir corrente contínua e alternada numa faixa de -100 A até +100 A. A relação entre a tensão medida pela porta analógica do Arduino e a corrente é de 20 mV/A, ou seja, 20 mV corresponde a 1 ampere. Na Figura 9 encontra-se uma ilustração do modelo do sensor utilizado.

Figura 11: Sensor de corrente ACS756SCA-100B-PFF-T, extraído do manual do mesmo

As correntes medidas serão negativa na faixa de 0 a $\frac{V_{cc}}{2}$ e positivas na faixa de $\frac{V_{CC}}{2}$ até V_{CC}. Note que devido a esta característica o valor medido está defasado de 512, que é metade de de 1024, definindo assim o zero da escala. O link do manual do sensor de corrente ACS756SCA-100B-PFF-T encontra-se nas referências. Desta forma, podemos estipular a seguinte equação:

$$I = \frac{V_{REF} \cdot (E_{A1} - 512)}{0,01} \tag{2}$$

No qual,
E_{A1} é a leitura analógica feita na porta A1 do Arduino;
I é a corrente medida em amperres.

5.3 Sensor de tensão

Utilizamos um divisor de tensão atuando como sensor de tensão, no qual a tensão de saída é dada pela seguinte equação:

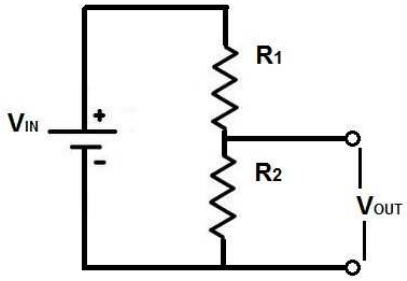

Figura 12: Divisor de tensão com sensor de tensão

Sendo assim o valor da Tensão de saída V_{out}, que é a tensão da bateria é dada por:

$$V_{out} = \frac{R_2}{R_1 + R_2} \tag{3}$$

Onde, No qual,
$R_1 = 1M\Omega$;
$R_2 = 100K\Omega$

Conforme o projeto de (ROSÁRIO,2013) e (SOUZA,2016), Foi utilizado um conversor conhecido com ponte H, pois seus conjunto de chaves se assemelham a letra "H", conforme mostrado na Figura 13. Dessa maneira, há três pontos distintos que a tensão deve ser medida. O pontos são na bateria e nos braços da ponte H, tornando possível a medição da tensão nos terminais do motor. A tensão na ponte H é dada por:

$$V_H = \frac{V_{REF} \cdot E_{A3}}{2^{10}} \tag{4}$$

Onde,
V_H é a tensão medida no braço da ponte H;
E_{A3} é a leitura analógica feita na porta A3 do Arduino.

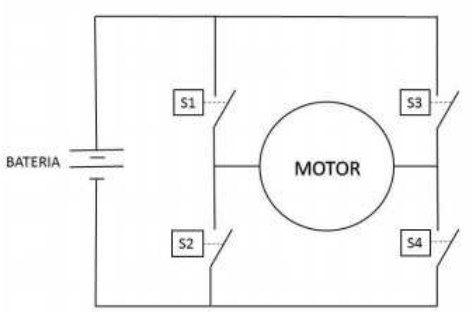

Figura 13: Ponte H

A medição da tensão da tensão no outro braço da ponte H é dada pela mesma equação 3, no entanto, a leitura analógica será feita na porta analógica A4.

6 Discussão dos Resultados

Todos os sensores utilizados neste projeto apresentaram uma casa decimal de confiabilidade, estando dentro das características desejadas para um bom funcionamento deste projeto. A página HTML recarregava os valores das variáveis de estado numa faixa de 2 a 3 segundos, o que foi também considerado suficiente para o funcionamento adequado do sistema de monitoramento e de um futuro sistema de controle e proteção.

7 Conclusão

Os componentes utilizados neste projeto são acessíveis quanto ao custo, o que permite a fácil manutenção dos mesmos. Este aspecto torna o projeto viável em termos financeiros (SOUZA, 2016). O Arduino Yún, apesar de apresentar pouco material de pesquisa, é configurado de maneira simples e apresenta uma boa confiabilidade. A Próxima etapa desse trabalho será a criação de um sistema de controle dos propulsores da embarcação e um sistema de proteção para o mesmo.

Referências

[1] ROSÁRIO, Rafael Vida de Castro (2013). *Sistema para Monitoração de uma Embarcação Não Tripulada*. Projeto de Graduação em Engenharia Eletrônica – UERJ. http://www.lee.uerj.br/ jpaulo/PG/2013/PG-Sistema-Monitoracao-Embarcacao-2013.pdf

[2] SOUZA, Lenielson Rodrigues, (2016). *Acionamento dos Motores CC de uma Embarcação Teleoperada*. Projeto de Graduação em Engenharia Eletrônica – UERJ. Dis-

ponível em: http://www.lee.eng.uerj.br/ jpaulo/PG/2016/PG-Acionamento-Motores-Embarcacao-2016.pdf

[3] CRUZ, Daiane Barbosa (2017). *Rede de Embarcações Não Tripuladas: Estudo do Arduino Yún para o Controle dos Propulsores*. Projeto de Iniciação Científica - UERJ. Disponível em: http://www.lee.eng.uerj.br/ jpaulo/PG/2017/Relatorio-Final-IC-2017-2017-Arduino-Yun.pdf

[4] Manual do LM35 Texas instruments. *Sensor de Temperatura*. Disponível em: http://www.ti.com/lit/ds/symlink/lm35.pdf

[5] Manuel do ACS756SCA-100B-PFF-T Allegro™. *Sensor de Corrente*.Disponível em: http://www.alldatasheet.com/datasheet-pdf/pdf/480257/ALLEGRO/ACS756SCA-100B-PFF-T.html

*Apêndice A- Algoritmo no Arduino

```
#include <Bridge.h>
#include <YunServer.h>
#include <YunClient.h>
BridgeServer server;

const int sensor_temperatura= A0;
const int test = A1;
float temperatura;
float tensao_bateria;
float corrente;
float a_temperatura = 5.0/(1023)/0.01;
float a_tensao = 5.0/1023*11;
float a_corrente = 5.0/(1023*0.02);
int b_corrente = 512;
void setup() {
  Bridge.begin();
  server.listenOnLocalhost();
  server.begin();
}
void loop (){
  BridgeClient client = server.accept();
  if (client) {
    String command = client.readString();
    command.trim();
    if (command == "temperature") {
      temperatura = float(analogRead(sensor_temperatura))*
         a_temperatura;
```

```
      client.print(temperatura);
      Serial.print("Temperatura: ");
Serial.println(temperatura);
    }
    if (command == "test") {
      temperatura = float(analogRead(test))* a_temperatura;
      tensao_bateria= float(analogRead(test))* a_tensao;
      corrente = float(analogRead(test)- b_corrente)* a_corrente;
      client.print("<p><h1>temperatura:");
      client.println (temperatura,1);
      client.println("</h1></p>");
      client.print("<p><h2>tens o na bateria:");
      client.println (tensao_bateria,2);
      client.println("</h2></p>");
      client.print("<p><h3>corrente:");
      client.println(corrente,3);
      client.println("</h3></p>");
      }
    client.stop();
  }
  delay(50);
}
```

*Apêndice B - Algoritmo da Página HTML

```

<!DOCTYPE html>
<head>
        <script type="text/javascript" src="zepto.min.js"></script>
    <script type="text/javascript">
        function refresh() {
            $('#content').load('/arduino/test');
                }
    </script>
</head>
<body onload="setInterval(refresh, 200);">
        <span id="content">Waiting for Arduino...</span>
</body>
```