

# Projeto e implementação de robô autônomo seguidor de linha

**Willian Americano Lopes**

Universidade Tecnológica Federal do Paraná - Câmpus Pato Branco  
DAINF - Departamento Acadêmico de Informática

Orientador: Professor Fábio Favarim  
Co-orientador: Professor César Rafael claire Torrico

walopes23@gmail.com  
{favarim,torrico} @utfpr.edu.br

05 de dezembro de 2017

# Sumário

- 1** Introdução
- 2** Objetivos
- 3** Metodologia
- 4** Justificativa
- 5** Materiais
- 6** Desenho e implementação
  - Projeto do *hardware*
  - Projeto do controlador de SED
  - Projeto do sistema de mapeamento
  - Função de transferência do veículo
  - Projeto do controlador de tempo contínuo
- 7** Resultados
  - Projeto e implementação da estrutura mecânica
  - Participação em competições
  - Teste dos sensores de refletância
- 8** Conclusão

- Industriais
- Médicos
- Móveis
  - Com pernas  
*(legged)*
  - Com rodas  
*(wheeled)*



(a)



(b)

Figura 1: Robôs móveis: (a) Robô Sojourner (NASA,1997);(b) Legged Squad Support System (DYNAMICS,2016)

# Controle de tempo contínuo

- Malha aberta
- Malha fechada
- Controlador
  - Proporcional
  - Integral
  - Derivativa

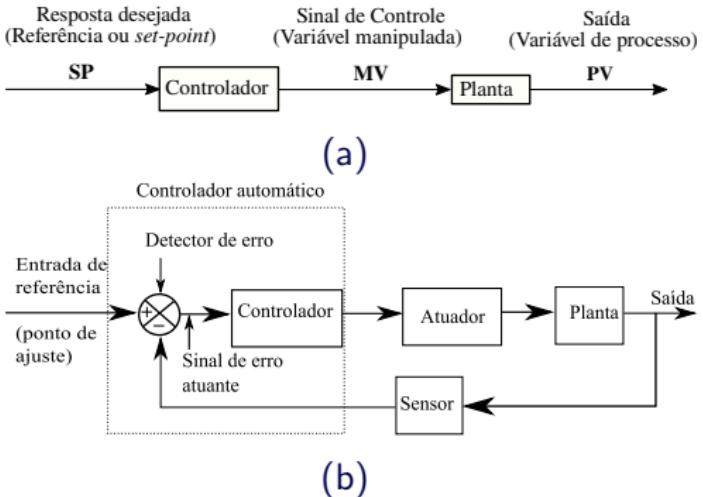


Figura 2: Diagramas de controle: (a) Malha aberta; (b) Malha fechada com controlador

# Controle a eventos discretos

- Eventos discretos
- Autômato
  - Moore
  - Mealy

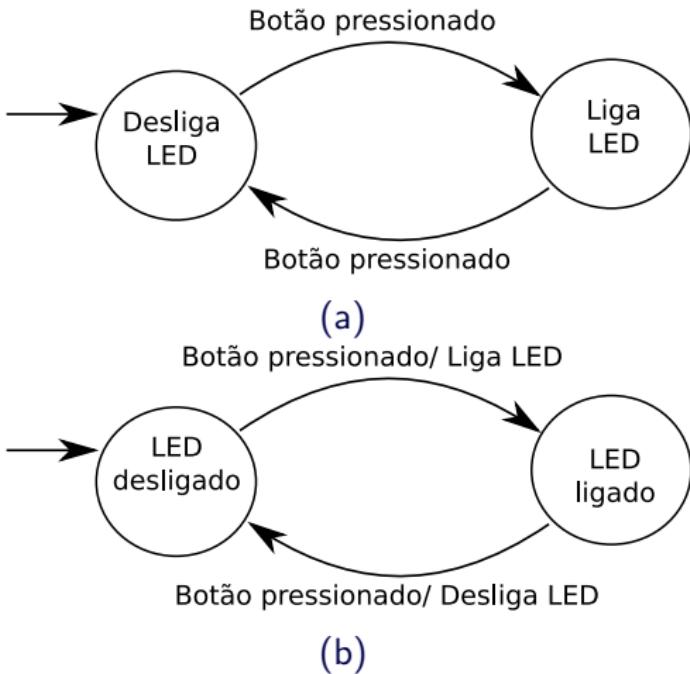
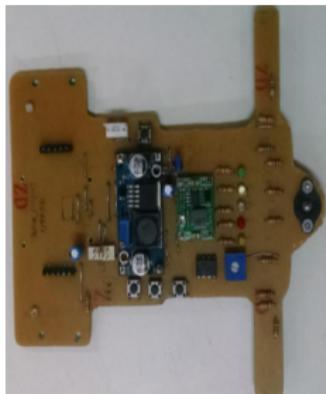


Figura 3: Autômatos: (a) Moore; (b) Mealy

# Trabalho de Petry (2016)

- Controle híbrido
- Controlador PID
- Lógica *fuzzy*



(a)



(b)

Figura 4: Robôs de Petry (2016): (a) Alpha project; (b) Pololu 3pi modificado

# Objetivos

## Geral

Projetar e implementar um protótipo de um robô seguidor de linha, que seja autônomo, através da utilização de controle híbrido, aperfeiçoando as técnicas desenvolvidas por Petry (2016).

## Específicos

- Projetar o condicionamento de sinais necessários para os dispositivos a serem utilizados, permitindo uma precisão na leitura dos sensores;
- Projetar e confeccionar a estrutura do protótipo, visando atender as dimensões especificadas pela Robocore;
- Implementar um controlador de Sistemas a Eventos Discretos, para que seja possível tratar de maneira precisa as marcações laterais da pista;

## Específicos

- Modelar a função de transferência do robô;
- Implementar um controlador para manter o robô sobre a linha na pista;
- Realizar testes com o protótipo em pistas que sigam as normas da Robocore;
- Fazer o mapeamento do percurso com um *encoder para o controle da velocidade*
- Comparar os resultados obtidos com o de Petry (2016).

# Metodologia

- 1 Definição e aquisição dos componentes;
- 2 Condicionamento de sinais dos sensores;
- 3 Projeto da estrutura mecânica e *software* embarcado;
- 4 Projeto do controlador;
- 5 Integração do sistema e implementação do protótipo;
- 6 Testes de desempenho;
- 7 Implementação do projeto final.

# Justificativa



(a)

(b)

Figura 5: Autômatos:  
(a) Moore; (b) Mealy

# Justificativa

ROBOGAMES

(a)



(b)

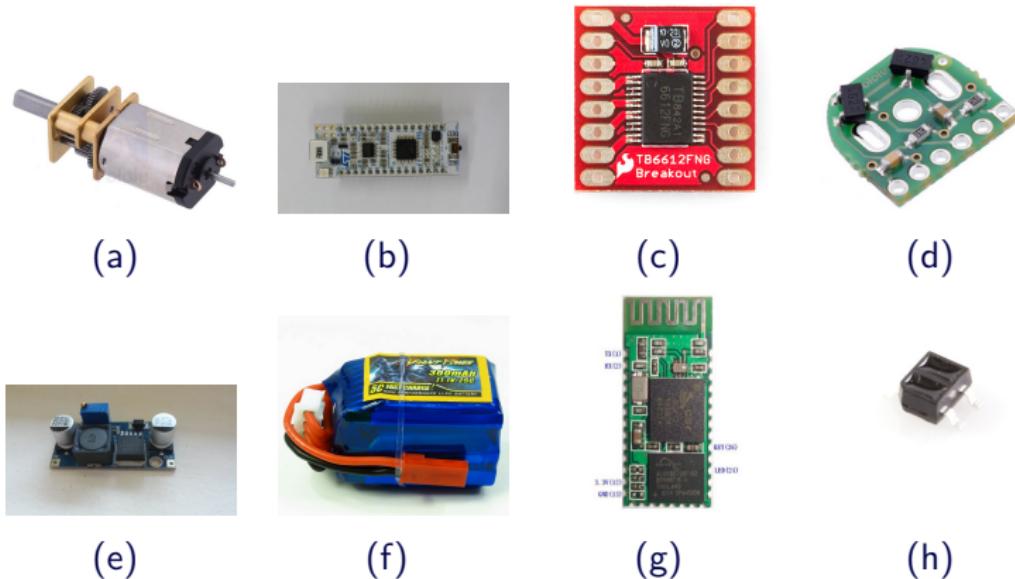
Figura 5: Autômatos:  
(a) Moore; (b) Mealy

- ART ( Autonomous Rail Rapid Transit ou Trilho Autônomo de Trânsito Rápido)
- Trem em Zhuzhou (China)



Figura 6: Trem sobre trilhos virtuais (DAILYMAIL, 2017)

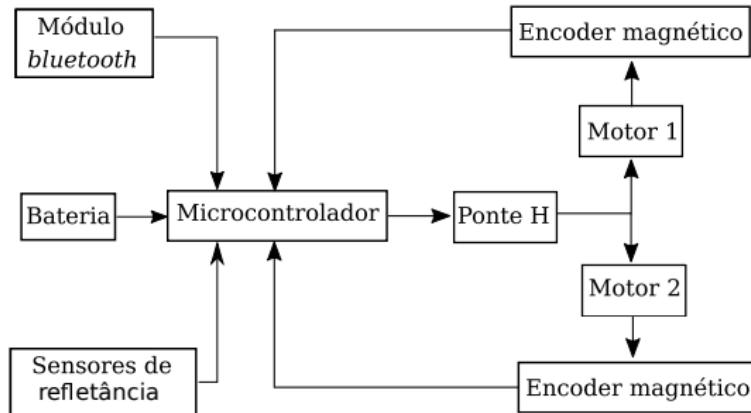
# Materiais



**Figura 7: Materiais utilizados: (a) Motor Pololu; (b) Nucleo-STM32F303K8; (c) TB6612FNG; (d) Encoder Pololu; (e) Step up; (f) Bateria LiPo; (g) HC05; (h) QRE1113**

# Desenho e implementação

## Projeto do *Hardware*



**Figura 8: Diagrama de funcionamento do hardware do veículo.**

# Sensores e condicionamento de sinais

## ■ QRE1113

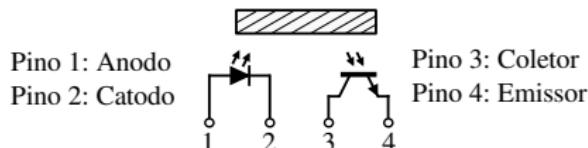


Figura 9: Configuração do sensor  
QRE1113.

# Sensores e condicionamento de sinais

## ■ QRE1113

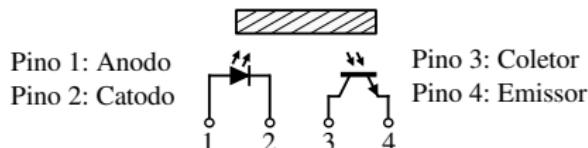


Figura 9: Configuração do sensor QRE1113.

## ■ AMS1117

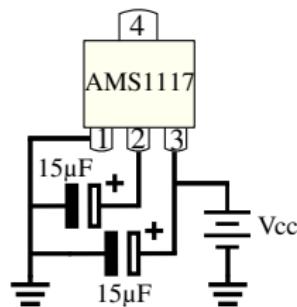
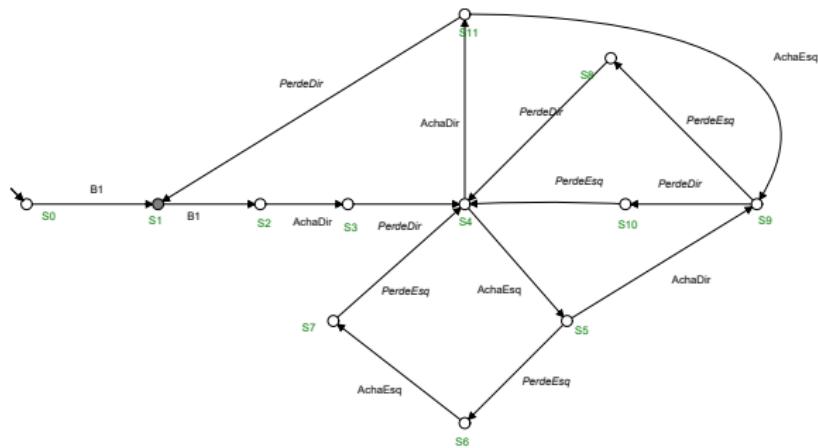


Figura 10: Configuração do regulador de tensão AMS1117

## Projeto do controlador de SED

- Modelado pelos softwares Supremica e Deslab
  - Autômato de Moore
  - Variáveis mapeadas:
    - B1 (botão);
    - AcharDir;
    - PerdeDir;
    - AchaEsq;
    - PerdeEsq.



**Figura 11: Controlador de SED**

# Projeto do sistema de mapeamento

- Armazenamento na memória *flash*
- Contém as seguintes informações:
  - Quantidade de marcações;
  - Início e fim de marcação da direita;
  - Início e fim de marcação da esquerda;

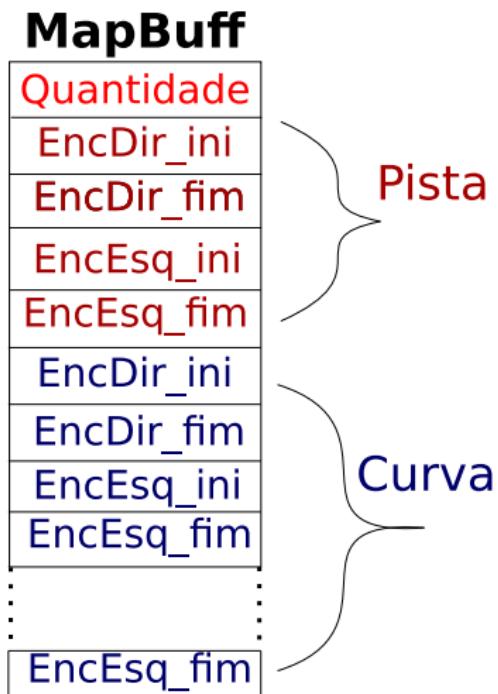


Figura 12: Vetor de armazenamento na flash

# Função de Transferência do veículo

## Aquisição dos valores da planta

- Obtenção da posição
- Mesmo PWM para os motores
- Motores em sentidos opostos
- Informações enviadas via *bluetooth*
- É esperado um gráfico próximo a uma rampa

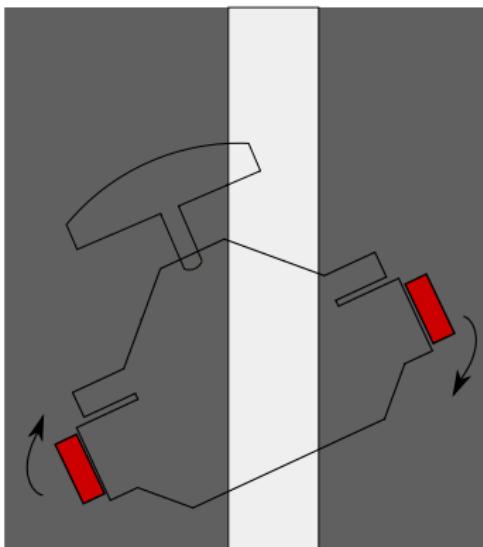


Figura 13: Aquisição dos valores de posição do robô

# Função de Transferência do veículo

## Modelo da função de transferência

Para encontrar a velocidade, deriva-se a posição

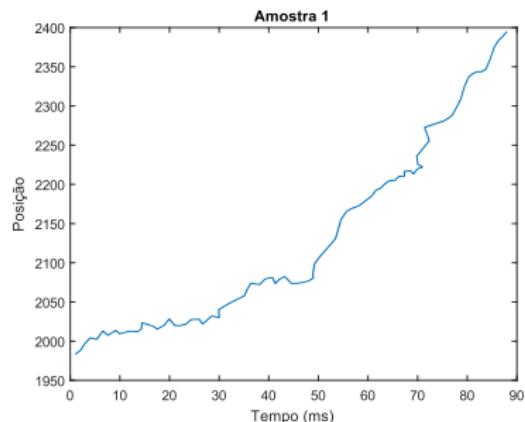


Figura 14: Gráfico da posição obtida

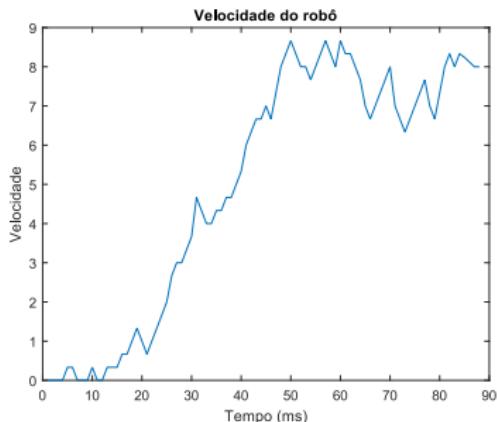


Figura 15: Gráfico da velocidade

# Função de Transferência do veículo

## Modelo da função de transferência

Com o Matlab, verificou-se uma função que mais se aproxima da desejada

- $P2ZU = 0,0054915 \left( \frac{1-0,011087s}{1+0,014889s+0,00023} \right)$
- A função contém 3 polos e 1 zero

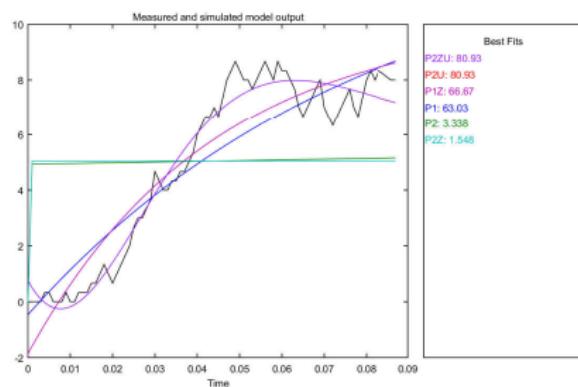


Figura 16: Comparativo feito pelo System Identification

## Modelo da função de transferência

- A função encontrada é relacionada à velocidade
- Para a original, é necessário integrá-la e multiplicá-la pelo ganho
- $FT : 1223\left(\frac{P2Z2}{s}\right)$

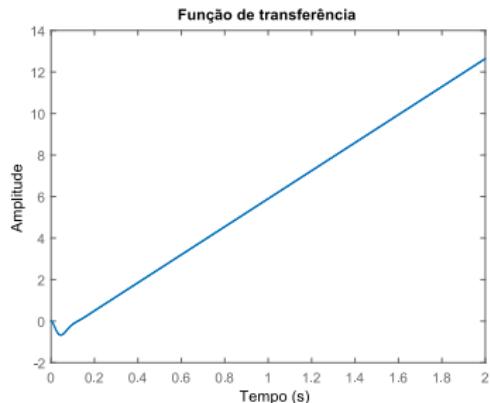


Figura 17: Função de transferência encontrada

# Projeto do controlador de tempo contínuo

- Controlador Proporcional-Derivativo (PD)
- Variável controlada é a posição

$$posicao = \frac{\sum_{n=1}^5 1000(n - 1)V_n}{\sum_{n=1}^5 V_n} \quad (1)$$

$$posicao = \frac{0 * V_1 + 1000 * V_2 + 2000 * V_3 + 3000 * V_4 + 4000 * V_5}{V_1 + V_2 + V_3 + V_4 + V_5} \quad (2)$$

# Projeto do controlador de tempo contínuo

## Trecho do código **pdcontrol()**

```
void pdcontrol(){...  
int16_t proporcional = posicao - 2000;  
// referencia e o valor 2000  
int16_t derivativo =  
(proporcional - proporcional_passado);  
//obtendo o derivativo  
proporcional_passado = proporcional;  
  
int16_t saida_pwm =  
(int16_t)((float)proporcional * Kp )  
+ ((float)derivativo * Kd );  
  
int16_t PWM = saida_pwm;
```

# Projeto do controlador de tempo contínuo

## Trecho do código **pdcontrol()**

```
if ( PWM > velocidade )    PWM =  velocidade;
//Saida limitada em 0 e 4095
if ( PWM < -velocidade )   PWM = -velocidade;
if (PWM < 0)
{
    __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_2,
    (velocidade+PWM));
    __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_4,
    velocidade);
}
if (PWM > 0)
{ ...
}
```

# Projeto e implementação da estrutura mecânica

- *Hardware* dividido em duas partes:
  - Barra frontal de sensores
  - Chassi (CrazyFrog)



Figura 18: Barra frontal de sensores

# Projeto e implementação da estrutura mecânica

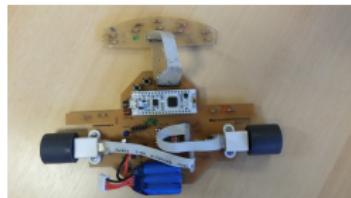


(a)

# Projeto e implementação da estrutura mecânica



(a)



(b)

# Projeto e implementação da estrutura mecânica



(a)



(b)



(c)

Figura 19: Chassis desenvolvidos: (a) CrazyFrog1; (b) CrazyFrog2; (c) CrazyFrog3

# Participação em competições

- Participação na FACE e WinterChallenge
- Competições disputadas com o CrazyFrog2
- Problema com os sensores
- Curva do 'S'



Figura 20: Pista do seguidor de linha Pro na WinterChallenge2017

# Teste dos sensores de refletância

- Sensores testados
  - QTR-1A, da Pololu;
  - QRE1113, da China;
  - QRE1113, da Arrow;
  - QRD1114.
- Procedimento para a aquisição

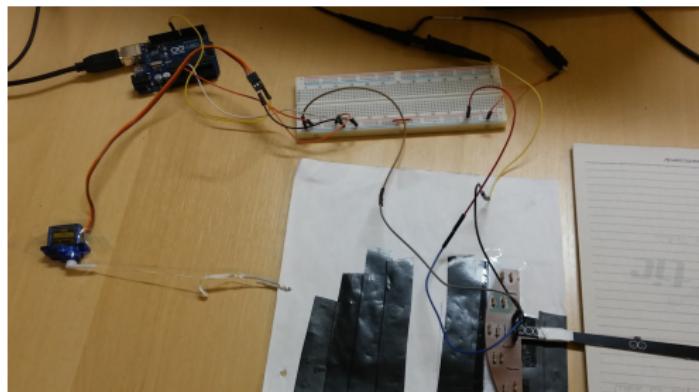
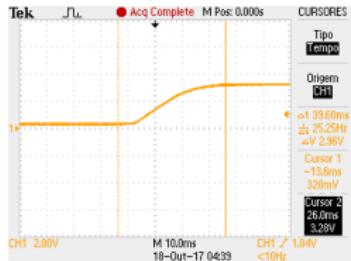
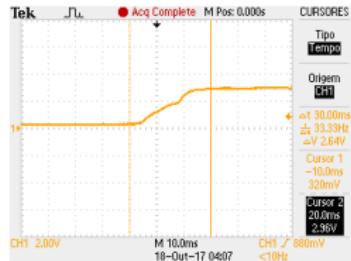


Figura 21: Mecanismo de aquisição do tempo de resposta dos sensores

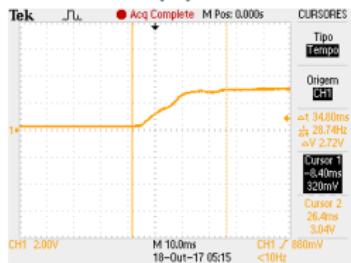
# Teste dos sensores de refletância



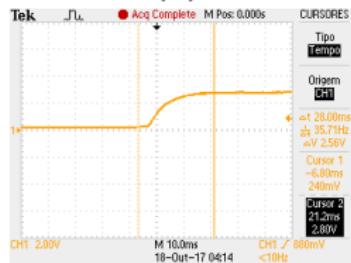
(a)



(b)



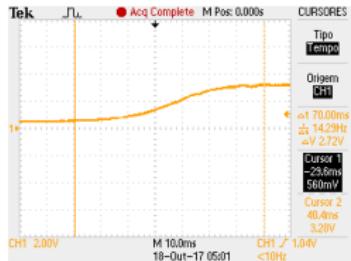
(c)



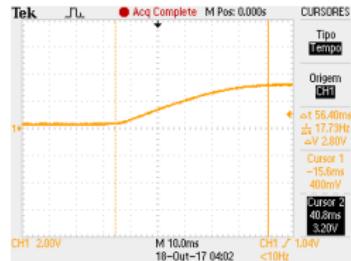
(d)

Figura 22: Teste dos sensores a 1mm de distância (a) QRE1113 do Aliexpress; (b) QRE1113 da Arrow; (c) QRD1114; (d) QTR-1A

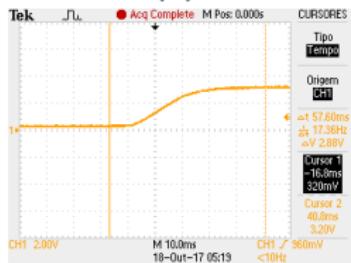
# Teste dos sensores de refletância



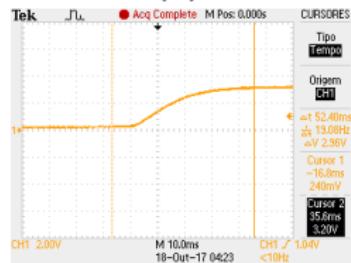
(a)



(b)



(c)



(d)

Figura 23: Resposta do sensores a 2 mm de distância: (a) QRE1113 do Aliexpress; (b) QRE1113 da Arrow; (c) QRD1114; (d) QTR-1A

# Conclusão

- Importância da participação em competições
- Qualidade inferior dos sensores utilizados
- Desempenho inferior ao esperado
- Melhorias no mapeamento
- Melhorias no controlador de SED

# Projeto e implementação de robô autônomo seguidor de linha

**Willian Americano Lopes**

Universidade Tecnológica Federal do Paraná - Câmpus Pato Branco  
DAINF - Departamento Acadêmico de Informática

Orientador: Professor Fábio Favarim  
Co-orientador: Professor César Rafael claire Torrico

walopes23@gmail.com  
{favarim,torrico} @utfpr.edu.br

05 de dezembro de 2017