

Confecção de um Shield para a Galileo Gen 2 para o Acionamento das Juntas e Leitura dos Sensores do Robô 2DSFJE

Arthur Ribeiro¹, Daniel Cunha¹, Matheus Tura¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{aribeiro, dmcunha, mttura}@inf.ufrgs.br

Resumo. *Este documento descreve o processo para a criação de um shield para a placa Intel Galileo Gen 2. Este hardware é capaz de acionar o motor e ler os sensores presentes no robô Quanser 2DSFJE. Ao longo do documento será descrito o processo de todas as etapas de desenvolvimento deste hardware, desde de sua etapa inicial de concepção até a etapa final de implementação.*

1. Introdução

O projeto proposto consiste no desenvolvimento e confecção de um shield para a Intel Galileo Gen 2, o shield deve ser capaz de acionar uma das juntas do robô Quanser 2DSFJE, a junta possui um motor com encoder, para o acionamento é necessário fornecer uma tensão entre -27 e 27 volts, o shield deve implementar uma ponte h que controla o valor médio desta tensão através de um sinal pwm. O ciclo de trabalho deste pwm representa o sentido e velocidade do movimento do motor, onde o ciclo de trabalho em 0% corresponde ao motor girando em velocidade máxima em um sentido, o ciclo de trabalho em 50% corresponde ao motor parado, o ciclo de trabalho em 100% corresponde ao motor girando em velocidade máxima no sentido contrário. O encoder do Quanser também deve ser decodificado pelo shield, para decodificar quadratura, foi utilizado um circuito integrado decodificador de quadratura, para que não haja erro na contagem e consequentemente dados inconsistentes.

Concomitantemente com o desenvolvimento do shield, deve-se desenvolver um software que implemente um controlador PID. O software deve incluir uma biblioteca com uma API. Com a API deve ser possível ler o valor do encoder em radianos, comandar o motor em Volts e a leitura do sensor de fim de curso. A biblioteca deve ser desenvolvida para executar no Linux como um programa no espaço de usuário, sem privilégios de superusuário. Os próximos capítulos detalharão o funcionamento do hardware e do software desenvolvido.

2. Hardware

Para possibilitar o controle a junta do Quanser utilizando a Galileo Gen 2, se desenvolveu um shield utilizando uma placa de circuito impresso. O shield possui dois circuitos distintos, um deles implementa uma ponte H que possui toda parte de alta potencia do shield, o outro circuito implementa o decodificador de quadratura.

Além disso, para maior parte do circuito foi optado por componentes SMD, devido ao seu menor tamanho e também menores problemas de mal contato, e praticidade na hora

da soldagem. O tamanho escolhido de SMDS para capacitores, resistores e LEDs foi de 0805, não foi escolhido abaixo deste tamanho, pois abaixo desse tamanho a dificuldade de solda poderia aumentar consideravelmente.

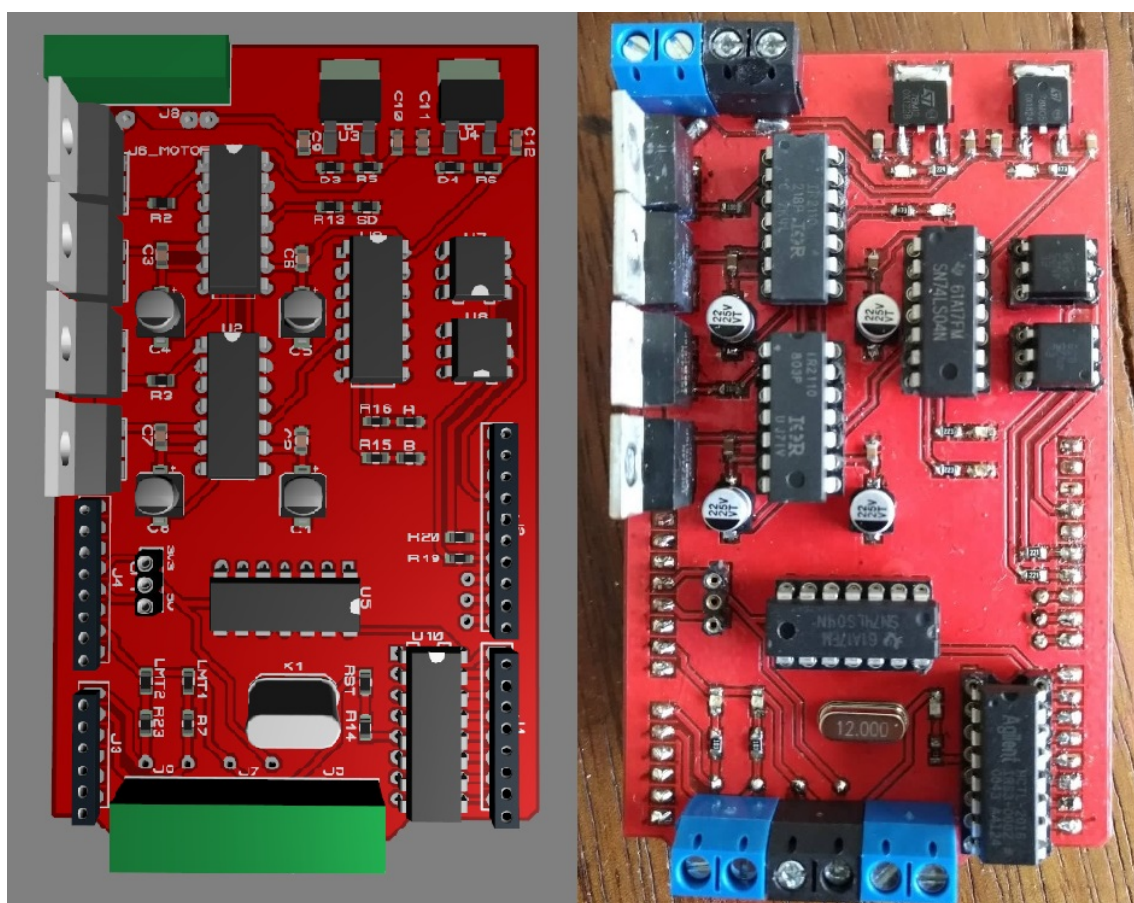


Figura 1. Ao lado esquerdo, simulação, ao direito, shield físico

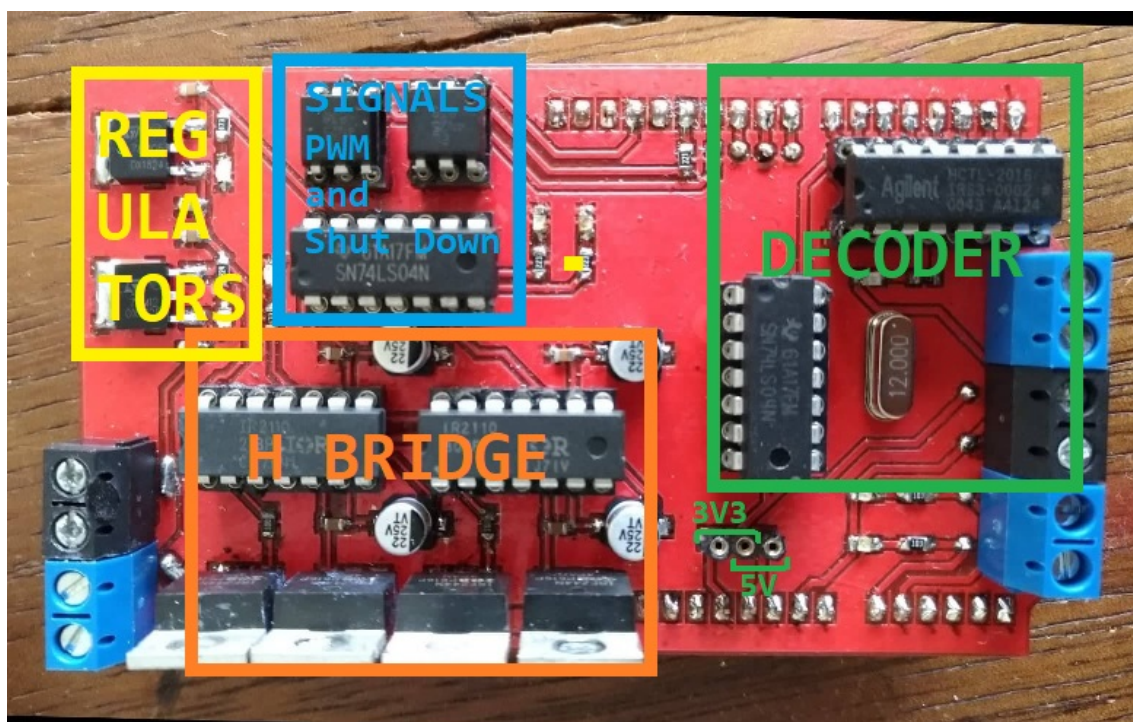


Figura 2. Blocos básicos

3. Alimentação e Reguladores

A Ponte H necessita de alimentação de 5V para controle lógico, e 12V para carregamento dos capacitores de bootstrap, para isso, um regulador LM7805 foi utilizado para gerar os 5V e um LM7812 para gerar os 12V. o Regulador LM7812 tem seu input conectado com o terminal de 27V da fonte Quanser, enquanto, para fins de reduzir a queda de tensão do LM7805, este tem seu input conectado na saída do LM7812. A corrente que passa LM7812 é a corrente dos capacitores, mais a corrente do LM7805, que tem sua maior potência gasta com LEDs. Logo, devido a baixa corrente, foi escolhido componentes reguladores em SMD, com sua dissipação no plano terra, visto que como receberiam pouca corrente, esquentariam pouco.

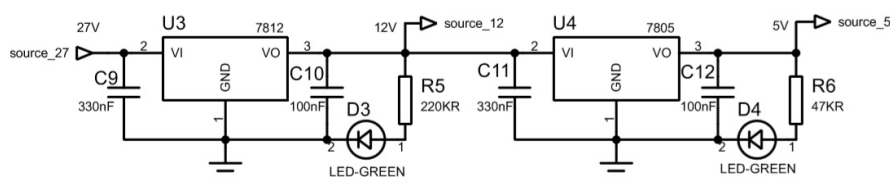


Figura 3. Reguladores de tensão 12V e 5V

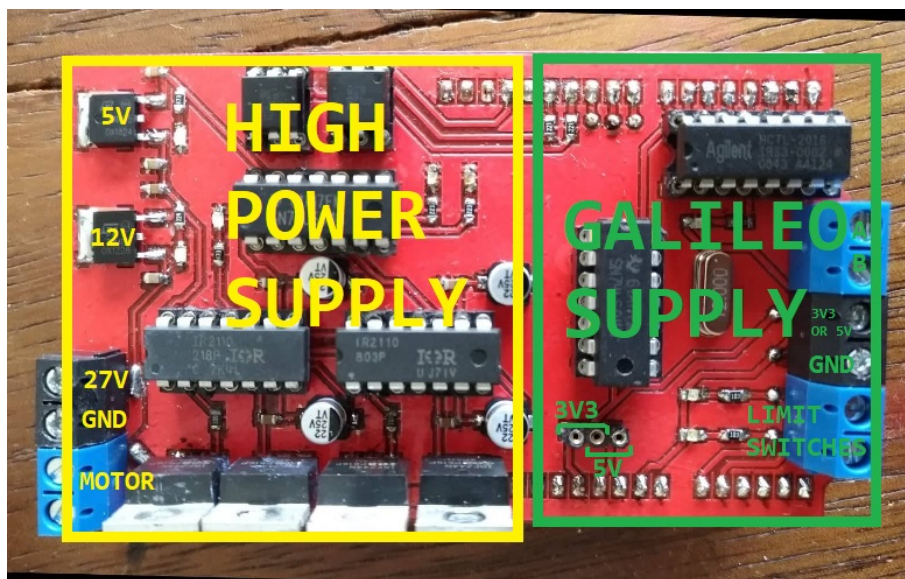


Figura 4. Separação da alimentação

4. Optoisolamento

Para que os sinais de controle da ponte H (pwm e shutdown) não tivessem sua alimentação misturada com a alimentação de potência, foram utilizados 2 optoacopladores 4N25 para isolar os dois circuitos, a imagem abaixo demonstra a separação do optoisolamento.



Figura 5. Imagem real dos optoacopladores nos sockets do shield

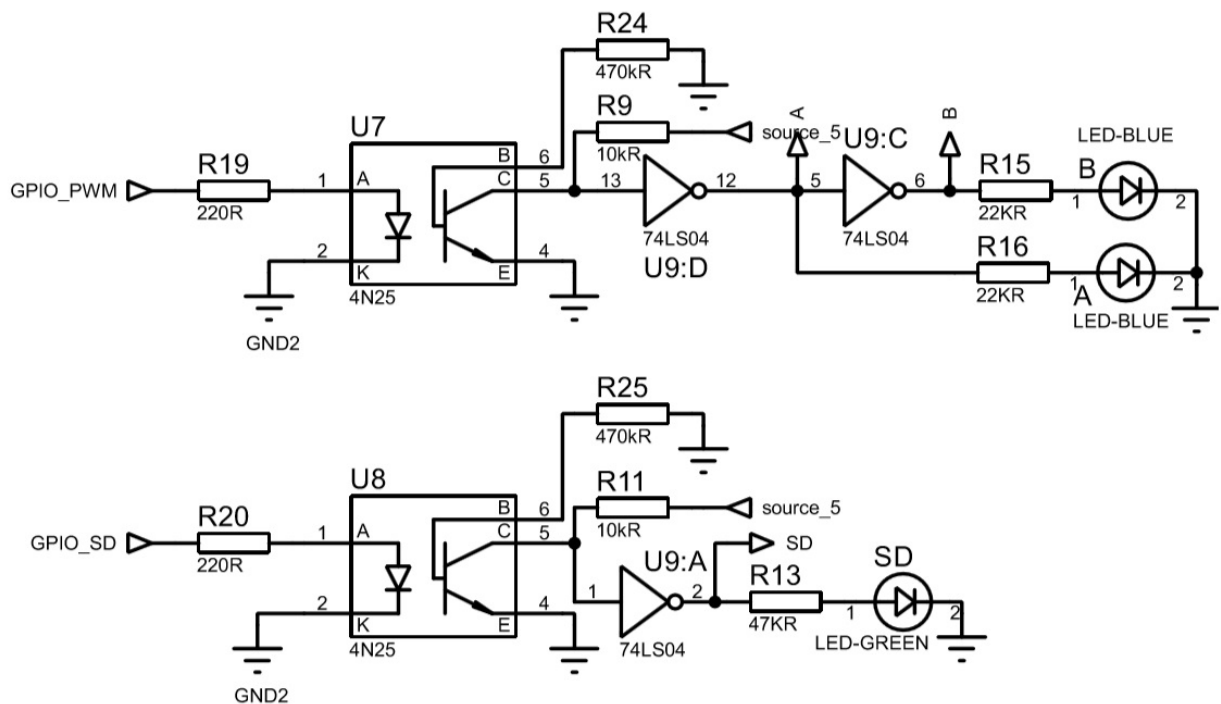


Figura 6. esquemático do circuito de opto-isolamento

5. Ponte H

Pelo fato do motor da junta Quanser ter um consumo de corrente máximo maior que 1A, o cuidado com a dissipação devido a potência dinâmica do chaveamento dos transistores de potência deve ser redobrado, para isso, os MOSFETs escolhidos para chavear a ponte H são todos do tipo NMOS, que naturalmente possuem velocidade de chaveamento maior que na família PMOS. Este modelo de ponte-H NMOS-only, possui menor potência dinâmica que uma ponte H convencional que utiliza PMOS na rede pull-up.

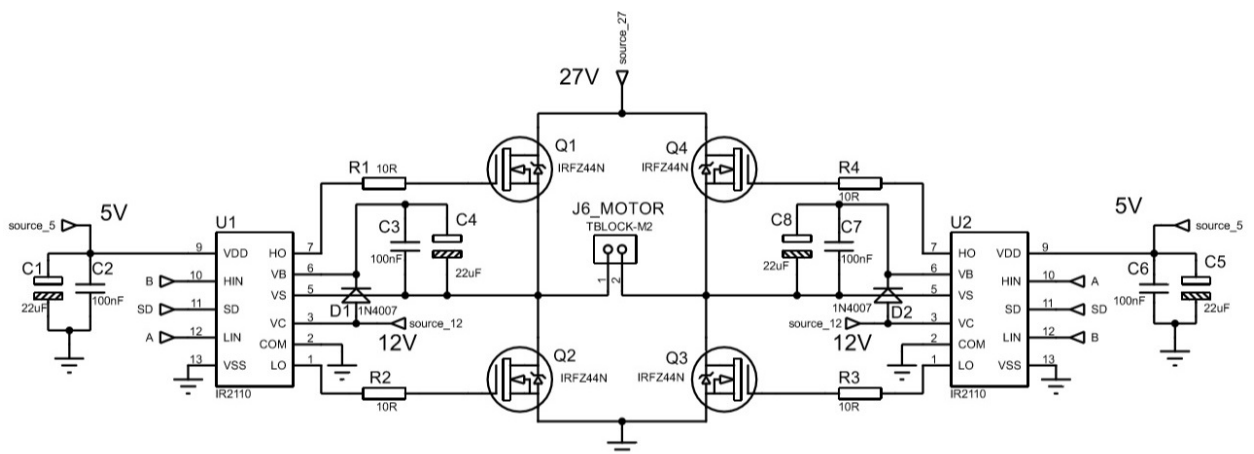


Figura 7. Ponte H NMOS

5.1. Capacitores de bootstrap

Para que um NMOS atinja a zona de saturação, a tensão V_{gs} deve ser maior que a tensão de threshold V_{th} satisfazendo as seguintes equações abaixo.

$$V_{gs} > V_{th}$$

$$V_{ds} > V_{gs} - V_{th}$$

Porém os transistores da rede pull-up em uma rede NMOS-only não acionam corretamente, visto que seus sources estão conectados nos terminais do motor, ao invés de GND. Esta tensão V_s , durante o acionamento dos gates, além de ser flutuante em alguns períodos, em outros, é uma tensão maior que o terminal de gate, não entrando na zona de saturação descrita na equação acima. Isso além de causar mal funcionamento, superaquece a rede pull-up, muitas vezes, queimando-a.

Para que o problema acima não ocorra e a tensão V_{gs} seja maior que a tensão V_{th} , é usada uma técnica de bootstrapping, onde são utilizados capacitores de 22uF em série com source dos MOSFETS da rede pull-up, conectado-os ao gate. A corrente de carregamento destes capacitores advém de de um regulador de 12V através de um diodo que garante a carga do capacitor, tais capacitores carregam uma tensão suficiente que somadas à tensão V_s de source, são mais que suficientes para ultrapassar o V_{th} , polarizando e saturando corretamente a rede pull-up durante o chaveamento.

Por fins de simplificação deste processo de bootstrap, o ciclo de carregamento e descarregamento destes capacitores é feito pelos CIs de bridge-driver IR2110, utilizados propriamente para isso.

6. Leitura de quadraturas do Decoder HTCL2016

O HTCL2016 é um encoder de 16 bits, mas que possui 8 pinos de saída para leitura, estes pinos são os bits do registrador/contador de quadraturas. Sua entrada de Clock é do tipo schmitt-trigger, e portanto com apenas um simples oscilador Pierce foi necessário para o correto funcionamento do decoder.

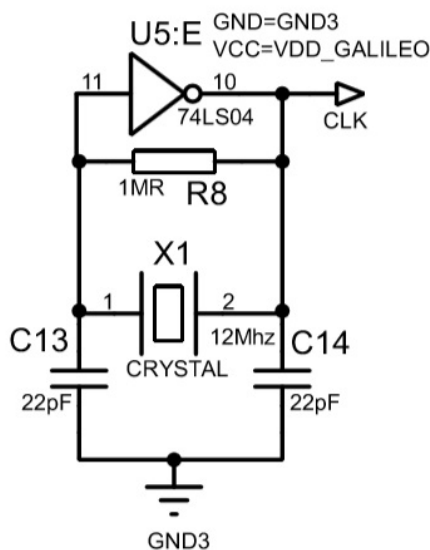


Figura 8. Esquemático do Oscilador Pierce

Além destes pinos o decoder, ele possui um pino de RST e um pino de Seleção(SEL) para selecionar entre o byte mais significativo ou menos significativo, porém para que a leitura seja feita corretamente, é necessário acionar na ordem correta o pino de !OE, que é responsável pelo acionamento dos buffer-tristates do registrador, permitindo que a leitura seja feita sem interferir na continua contagem das quadraturas.

Abaixo, o diagrama de pulsos de sinais que mostram a ordem correta, como também o tempo necessário entre um edge e outro, para a correta leitura.

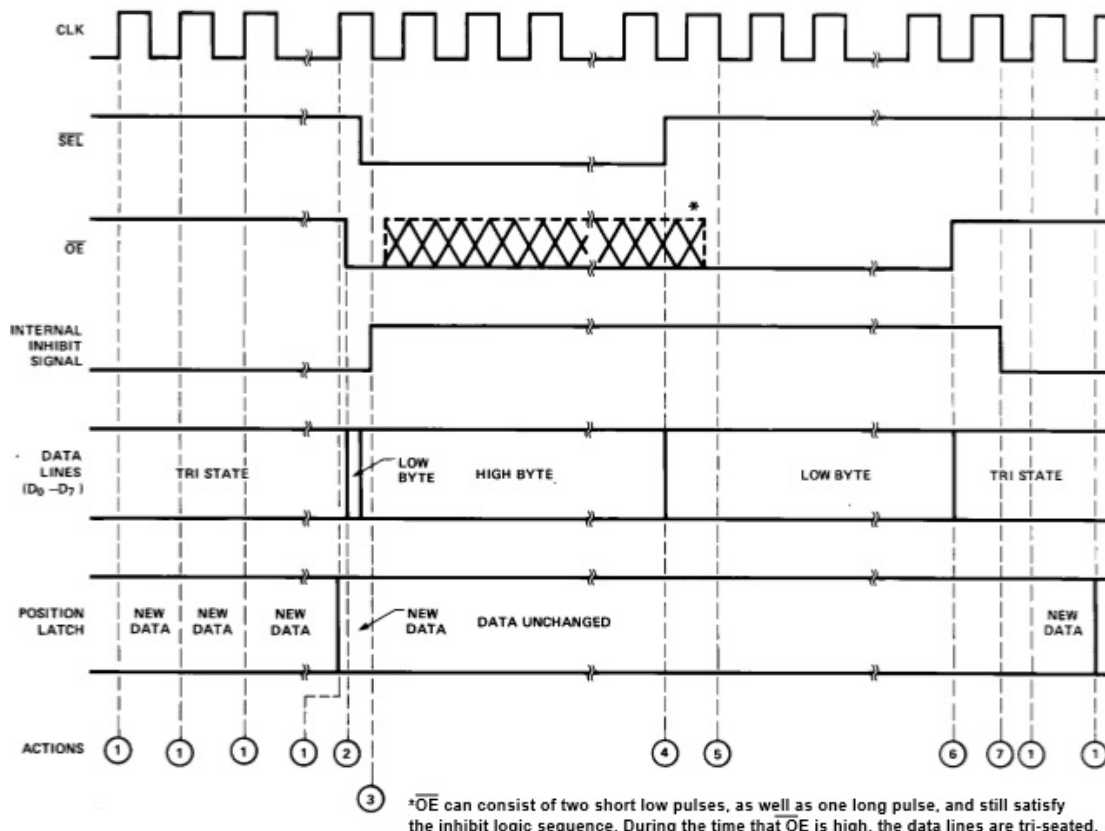


Figura 9. Diagrama de pulsos de sinais original do Datasheet

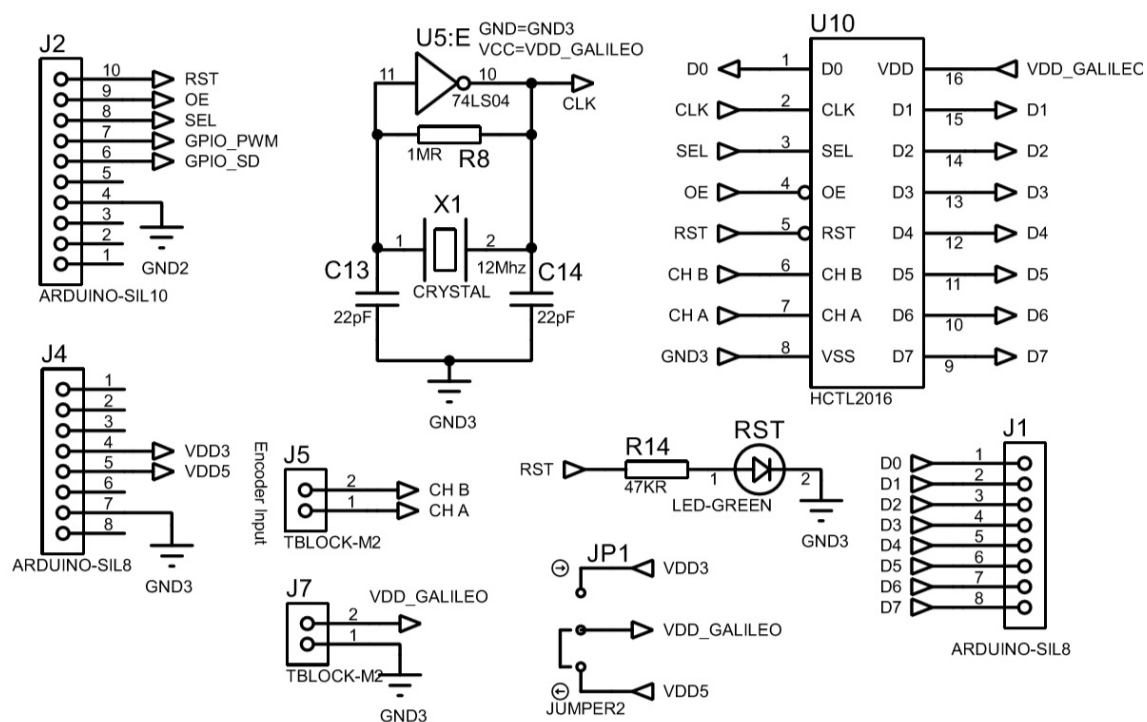


Figura 10. circuito de leitura de quadratura

7. Placa de Circuito Impresso

A placa de Circuito impresso, foi fresada no LASCAR, onde a CNC foi operada por Marcos Vizzotto.

Para obter a PCI sem erros, foram necessários 3 projetos físicos de PCI, visto que o projeto das duas primeiras eram mais difíceis de soldar e a disposição dos componentes não era confortável. Abaixo estão o top layer e bottom layer, como também o diagrama colorido, onde azul representa a layer bottom e a vermelha, a layer top.

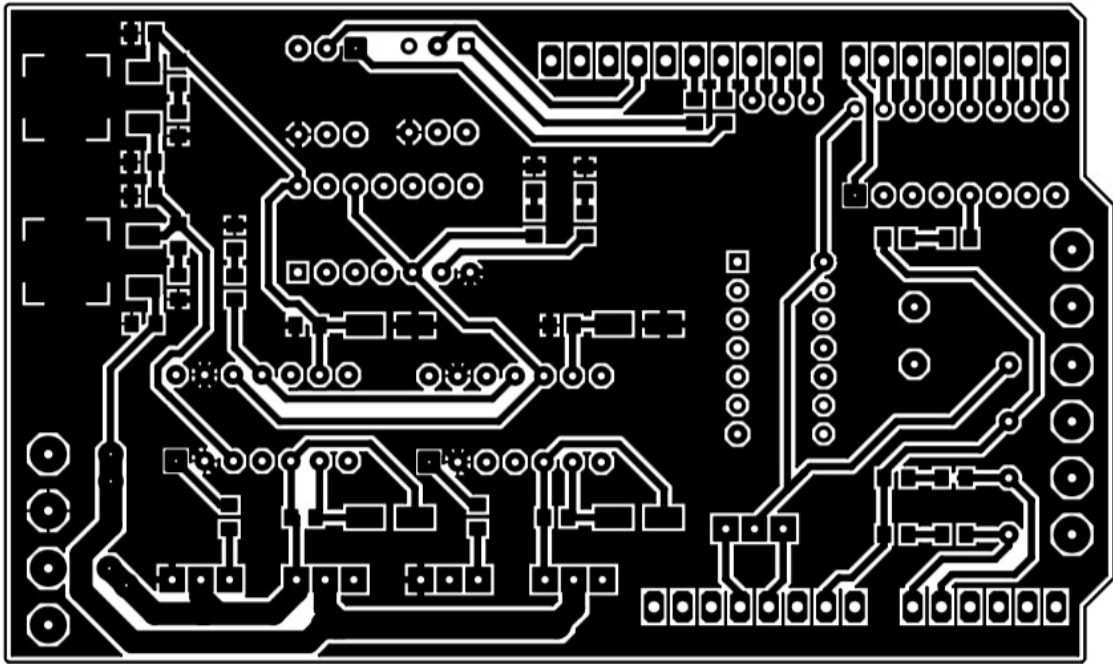


Figura 11. Top Layer

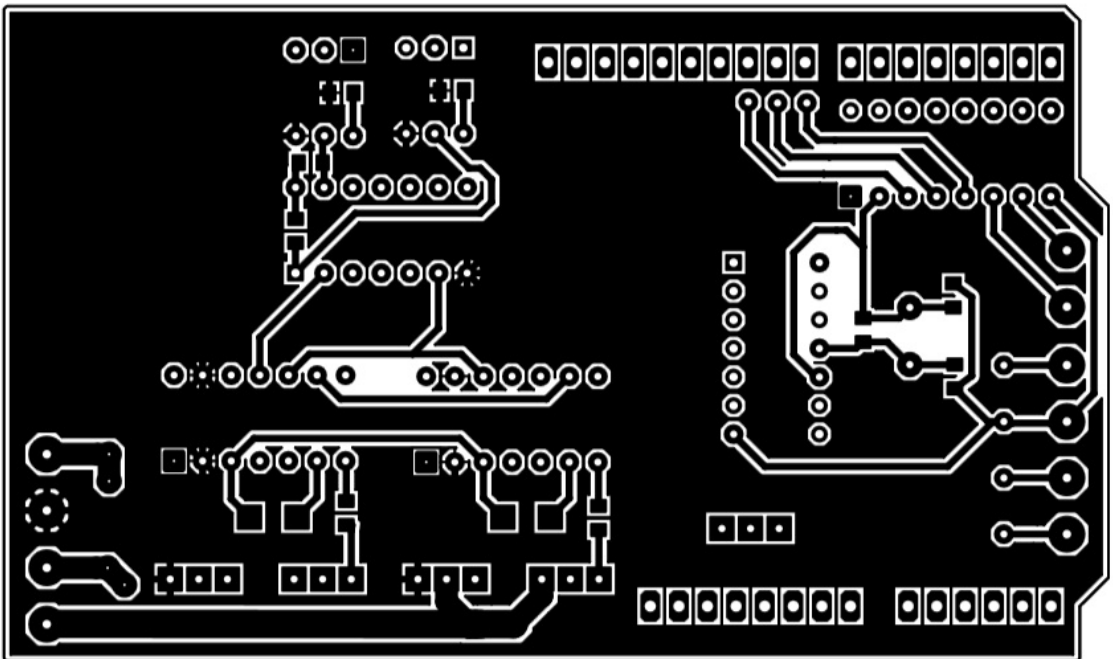


Figura 12. Bot Layer

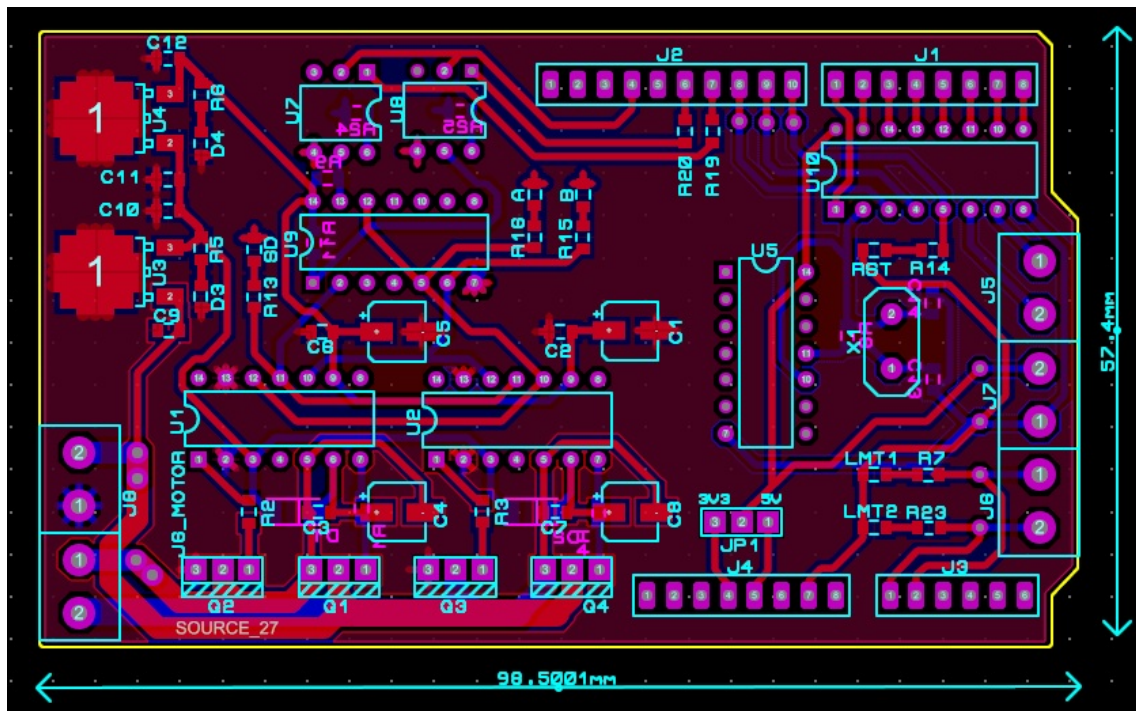


Figura 13. Todas camadas

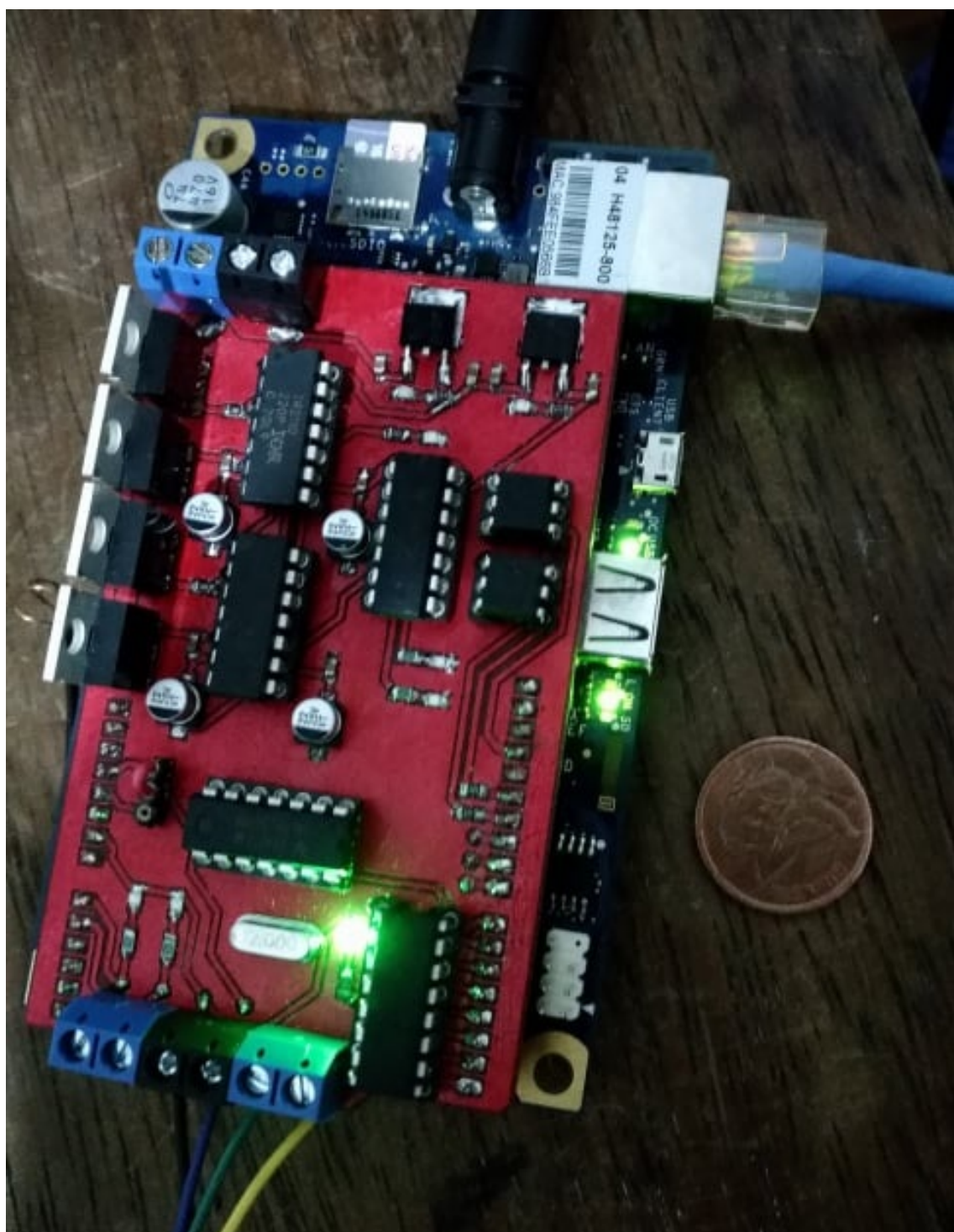


Figura 14. Placa montada, em teste de decoder, sem alimentação de potência