UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO

SISTEMAS DIGITAIS

Exame 2

JOSÉ TOBIAS SOUZA DOS SANTOS

LUCAS PEREIRA WANDERLEY DE OLIVEIRA

MARCUS PAULO SOARES DANTAS

RODRIGO DE AZEVEDO FERNANDES

NATAL/RN

2018

* **Questão 1**

Código Galileo:

import mraa, random #Importacao dos modulos

d3 = mraa.Pwm(3) #Definindo o pino digital 3 como saida PWM

d3.period\_us(700) #Definindo a frequencia da taxa de atualizacao da saida PWM

a0 = mraa.Aio(0) #Marcando o pino A0 como entrada analogica

a1 = mraa.Aio(1) #Marcando o pino A1 como entrada analogica

a2 = mraa.Aio(2) #Marcando o pino A2 como entrada analogica

a3 = mraa.Aio(3) #Marcando o pino A3 como entrada analogica

a4 = mraa.Aio(4) #Marcando o pino A4 como entrada analogica

d3.enable(True) #Ativando a saida PWM

d3.write(0) #Escrevendo 0 no pino D3

while True:

soma = 0

for i in range(5): #Laco for para ler todas as 5 entradas

if(i == 0): #Codigo para ler uma entrada analogica por loop do laco,

x = a0.read() # multiplicar com um valor aleatorio w e somar com a variavel

w = random.random() # soma

elif(i == 1):

x = a1.read()

w = random.random()

elif(i == 2):

x = a2.read()

w = random.random()

elif(i == 3):

x = a3.read()

w = random.random()

else:

x = a4.read()

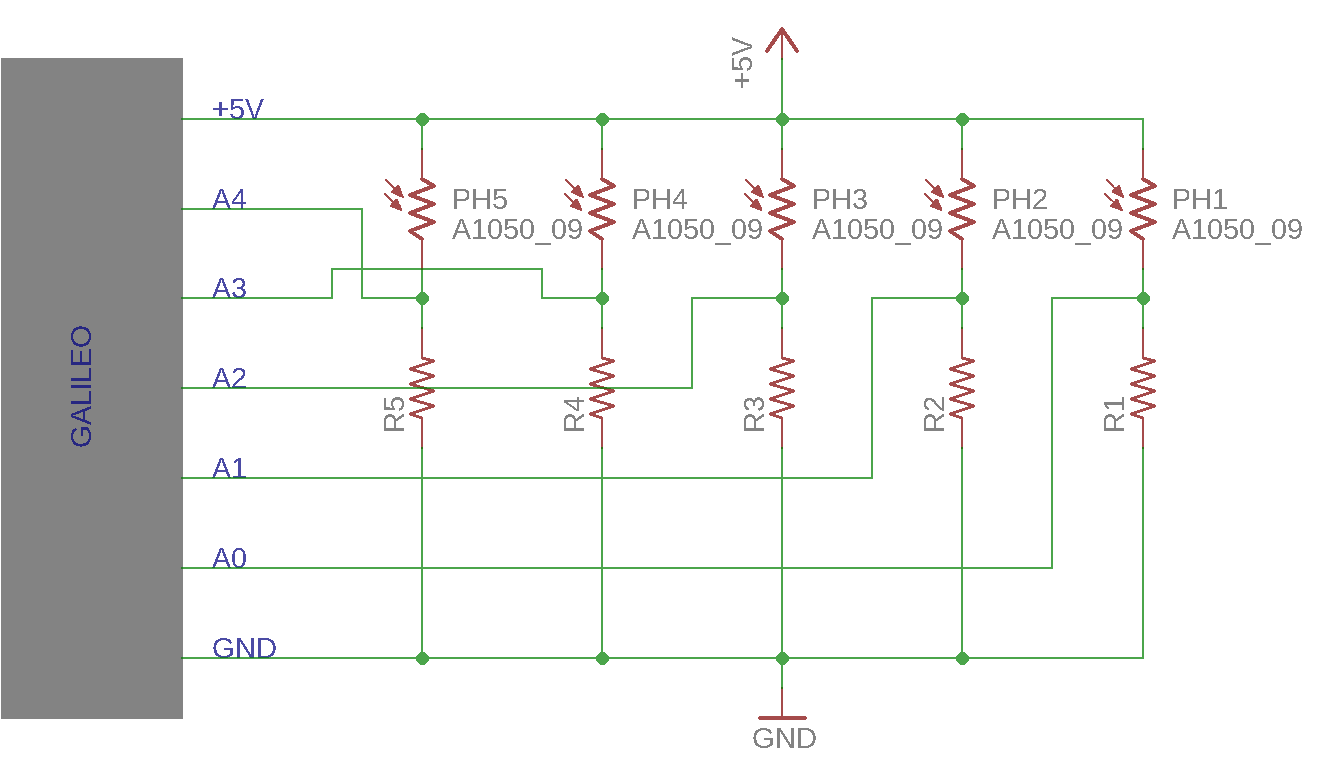
w = random.random()

soma = soma + w\*x

y = *int*(soma) #Separacao da parte inteira da soma

d3.write(y) #Escrita na saida PWM o valor de y

Esquemático:



* **Questão 2**

Código Galileo:

import mraa, time #Importando os modulos

spi = mraa.Spi(0) #Habilitando o SPI

spi.frequency(4000000) #Definindo a frequencia de 4MHz

tx = *bytearray*(3) #Criando o bytearray que armazenara os

tx[0] = 0 # valores dos bytes enviados

tx[1] = 0

tx[2] = 0

global i #Inicializando a variavel que contara de

i = 0 # forma crescente

while True:

i = i + 1

tx[0] = i #Definindo o valor de saida

print('Valor enviado: %d' % tx[0]) #Exibindo a saida antes do envio

rx = spi.write(tx) #Enviando os bytes

print('Valor recebido: %d' % rx[2]) #Exibindo o valor recebido

time.sleep(1)

Código Arduinos:

#include <SPI.h> //Importando a biblioteca SPI

byte rx = 0; //Definindo a variavel global que lidara

// com a entrada e a saida

*void* SlaveInit(*void*) { //Funcao responsavel por setar os pinos da

pinMode(SCK, *INPUT*); // comunicacao SPI

pinMode(MOSI, *INPUT*);

pinMode(MISO, *INPUT*);

pinMode(SS, *INPUT*);

SPCR = (1 << SPE); //Definindo o Arduino como slave

}

byte SPItransfer(byte value) { //Funcao responsavel por receber e enviar

SPDR = value;

while(!(SPSR & (1<<SPIF)));

delay(10);

return SPDR;

}

*void* setup() {

*Serial*.begin(9600);

SlaveInit(); //Inicializacao do bus SPI

*Serial*.println("Iniciando Slave");

}

*void* loop() {

if(!digitalRead(SS)){ //Se houver intencao de envio do mestre...

pinMode(MISO, *OUTPUT*); //... habilitar a entrada de dados...

*Serial*.println("Slave Ativado");

*Serial*.println("tx:" + String(rx)); //... imprimir o que sera enviado...

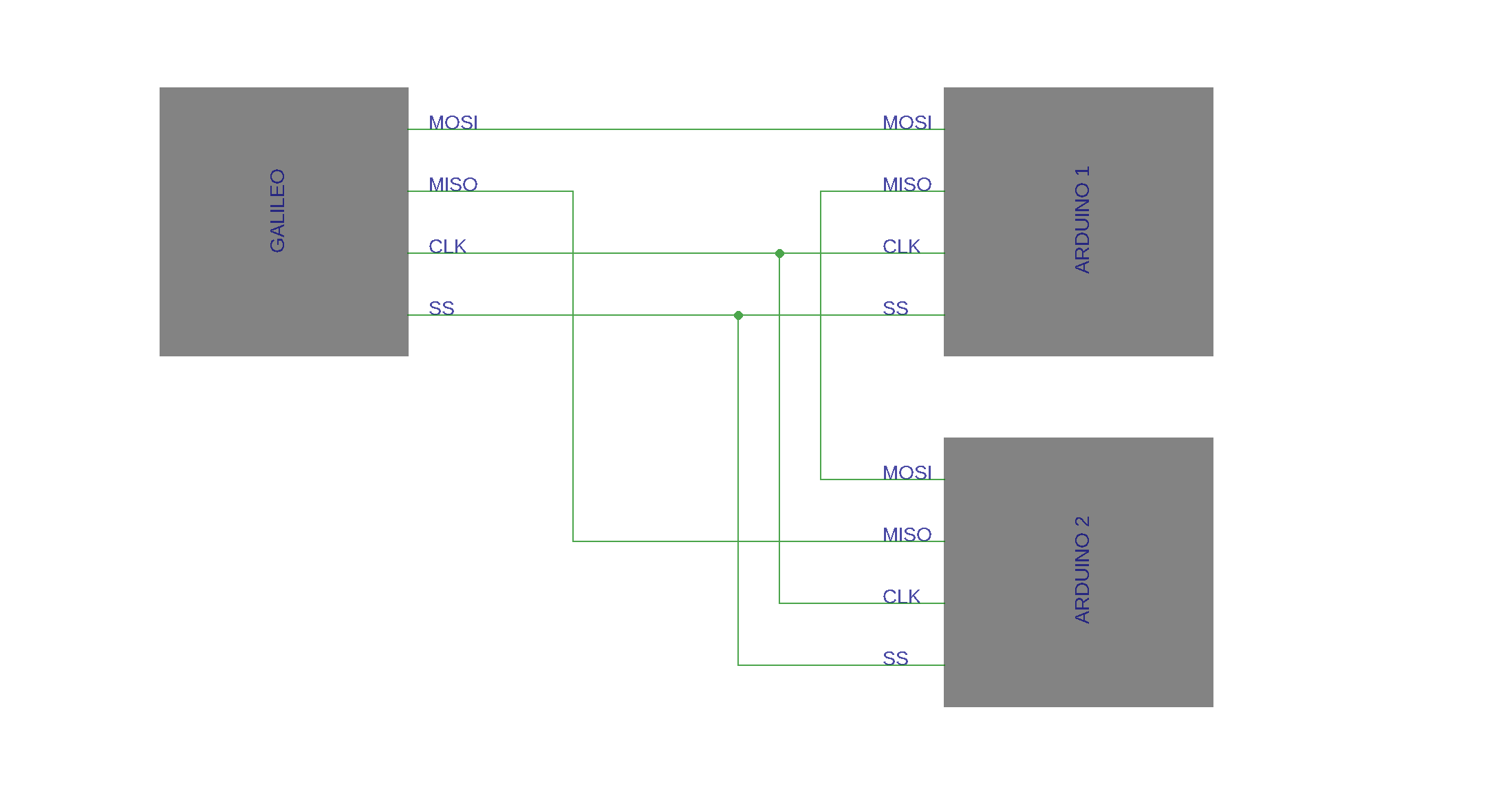
rx = SPItransfer(rx); //... enviar o valor existente na variavel rx e salvar o recebido...

*Serial*.println("rx:" + String(rx)); //... imprimir o valor recebido

}

}

Esquemático:



* **Questão 3**

Código Galileo:

import mraa #Importando os modulos

import time

u = mraa.Uart(0) #Inicializacao da UART

u.setBaudRate(115200) #Baudrate 115200

u.setMode(8, mraa.UART\_PARITY\_NONE, 1) #8 bits de dados, 1 de parada e sem paridade

u.setFlowcontrol(False, False) #Sem flow control

msg\_b = *bytearray*("Ola, Arduino em bytearray!", "ascii") #Enviando como bytearray

print("Enviando como bytearray: '{0}'".format(msg\_b))

u.write(msg\_b)

u.flush()

time.sleep(1.5) #Sleep para garantir o recebimento no Arduino

msg\_s = "Ola, Arduino em string!"

print("Enviando mensagem como string: '{0}'".format(msg\_s))

u.writeStr(msg\_s)

time.sleep(1.5)

u.writeStr("X")

print("Existe algo do outro lado?")

time.sleep(1.5)

if u.dataAvailable(100): #Esperando resposta, com 100ms de timeout

print("'{0}', diz o Arduino".format(u.readStr(20)))

else:

print("Nenhum dado recebido. Existe algo conectado?")

Código Arduino:

#include <SoftwareSerial.h> //Importando a biblioteca

SoftwareSerial mySerial(10, 11); // 10 = RX, 11 = TX

*void* setup() {

*Serial*.begin(115200);

while (!*Serial*) {

;

}

mySerial.begin(115200); //Inicializando a serial

}

*void* loop() {

if (mySerial.available()) { //Caso haja envio da Galileo

String rx = mySerial.readString();

*Serial*.println(rx); //Exibindo o texto na tela

if(rx == "X"){ //Caso venha um X...

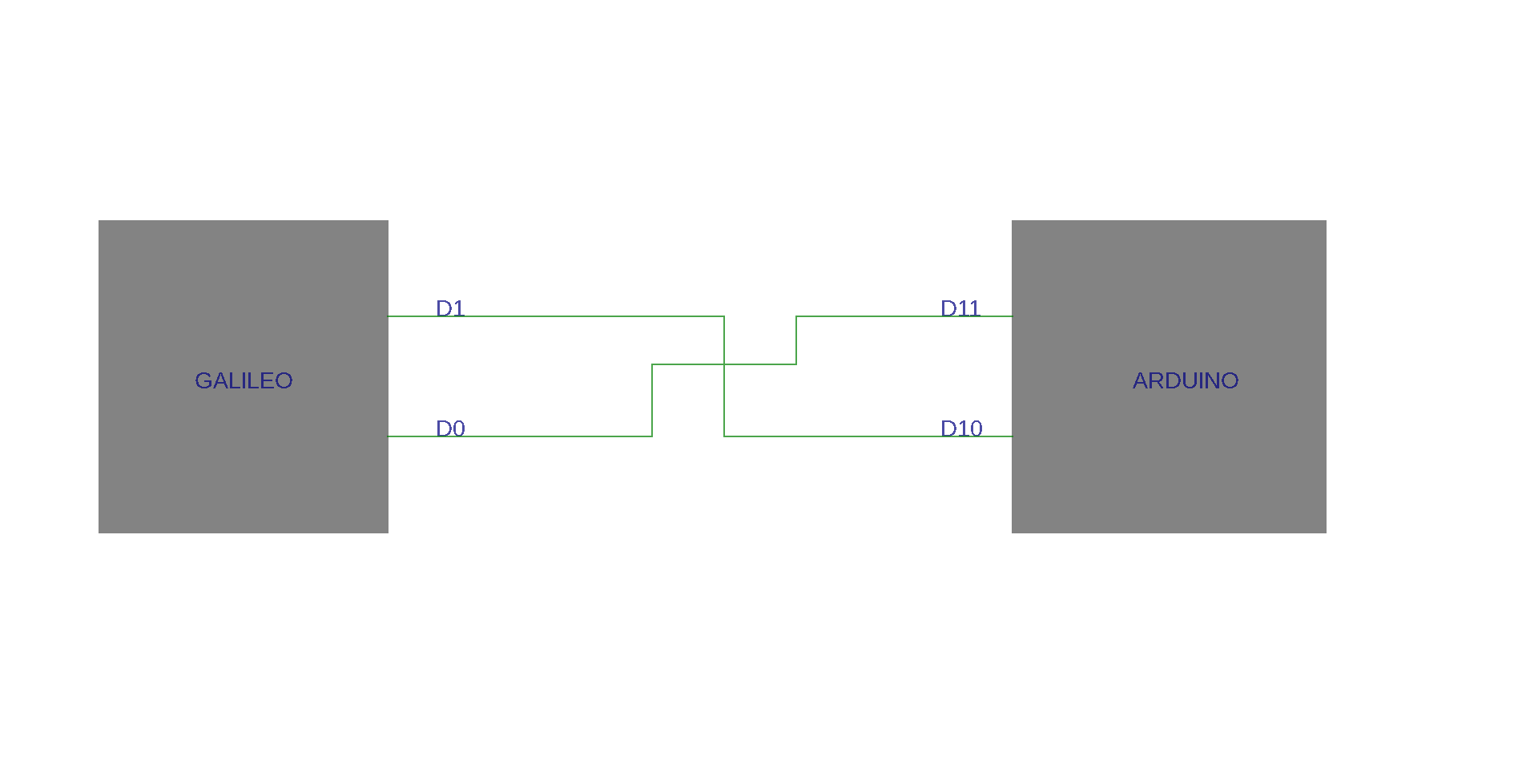
mySerial.write("Sim, mestre!"); //... envie uma string especial

}

}

}

Esquemático:



* **Questão 4**

Realizar o proposto na Questão 1 com um processador de uso geral como o encontrado na Galileo pode ser vantajoso pela possibilidade de utilizar diversas linguagens de programação, pela rápida forma de acesso às entradas analógicas do sistema, pela capacidade de conexão com a Internet e pela possibilidade de realizar diversas outras operações enquanto executa o código. Como desvantagem, há o custo elevado, seu tamanho físico, o consumo de energia elétrica, em comparação com um MCU e o potencial desperdício de capacidade de processamento, caso sua proposta seja única e exclusivamente a relatada na Questão 1. Já um MCU possui a vantagem de ser menor, com a possibilidade de ser embarcado em um produto pequeno, o baixo consumo de energia elétrica, o custo infinitamente menor e um processamento rápido, para o que foi proposto na questão. Como desvantagens, possui velocidade de processamento reduzida, poucas linguagens de programação possíveis de utilizar, não possui conexão à Internet embutida e uma possibilidade de processamento paralelo bastante limitada.