Universidad de Costa Rica Elena Cantero-Acosta (B81583)

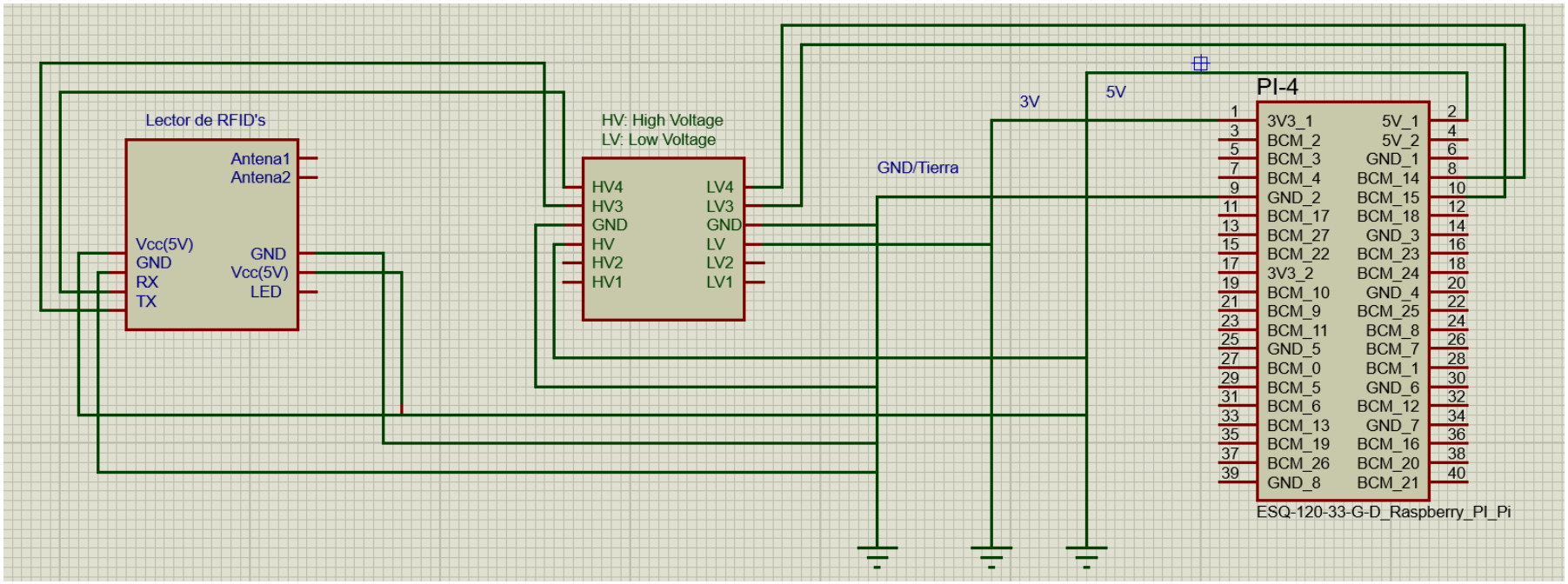
Escuela de Biología elena.cantero@ucr.ac.cr

Manual Circuito Básico

Diagrama del circuito

Es importante tomar en consideración el siguiente esquema que muestra la disposición física de los pines de una Raspberry Pi 4, las cuales se pueden leer en las descripciones de los pines del diagrama de circuito de Proteus. Además se adjunta una tabla con las conexiones entre los pines de cada circuito integrado para facilitar la lectura.

Conexión 1	Terminal
Lector de RFID's	
Antena 1	Antena externa cable 1
Antena 2	Antena externa cable 2
GND	Tierra común
Vcc(5V)	Alimentación de 5V
LED	----
Vcc(5V)	Alimentación de 5V
GND	Tierra común
TX	HV4
RX	HV3
Raspberry Pi 4	
8	LV4
10	LV3
9	Tierra común
1	Alimentación de 3V
2	Alimentación de 5V
Nivelador de tensión eléctrica	
GND (ambas)	Tierra común
LV	Alimentación de 3V
HV	Alimentación de 5V



Se tienen 3 Circuitos integrados, de izquierda a derecha serían: el lector de RFID's, el nivelador y la Raspberry Pi 4. De momento, se trabajará en una protoboard para poder desmantelar el circuito y utilizar los componentes a conveniencia. La protoboard cuenta con unas columnas largas y filas anchas separadas entre sí, cada columna larga o fila ancha está unida entre sí y representa una misma terminal o nodo. En las columnas largas se acostumbra colocar las tierras y las alimentaciones para poder alimentar el circuito desde diferentes puntos, allí se van a colocar una tierra común en una columna, la alimentación de 3V en otra columna y los 5V en otra. Tal y como puede verse en el diagrama de circuito. Todas diferentes entre sí porque sino se tendría un corto circuito y todo lo que esté conectado se quemaría.

La Raspberry Pi envía y recibe señales en los pines que se manejan como un 1 o 0 lógicos, con 3V o 0V asociados. Sin embargo, el lector de RFID's trabaja con señales digitales de 5V y 0V, respectivamente. Para lograr una adecuada comunicación entre ambos dispositivos es necesario utilizar un nivelador de tensión eléctrica como el que se muestra en el circuito. Éste debe estar alimentado con las dos tensiones asociadas al 1 lógico (3V para el Low Voltage y 5V para el High Voltage) y una tierra común; la cuál debe ser la misma para ambas tensiones del nivelador, la Pi y el lector de RFID's. Gracias a la versatilidad que tiene la Raspberry Pi 4, su diseño incluye algunos puertos dados únicamente para alimentación, tanto de 3V como de 5V; por lo que no hace falta una batería o alimentación externa.

Código

Código base sobre el cuál se trabajó y se hicieron mejoras hasta obtener el funcionamiento adecuado:

PYTHON CODE

We wrote the following simple code and tried to read some tags.

```
import time
import serial
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)

Tag1 = str('1800387799CE')
GPIO.setup(23,GPIO.OUT)
GPIO.setup(24,GPIO.OUT)
GPIO.output(23,False)
GPIO.output(24,False)
PortRF = serial.Serial('/dev/ttyAMA0',9600)
while True:
    ID = ""
    read_byte = PortRF.read()
    if read_byte=="\x02":
        for Counter in range(12):
            read_byte=PortRF.read()
            ID = ID + str(read_byte)
            print hex(ord( read_byte))
        print ID
        if ID == Tag1:
            print "matched"
            GPIO.output(23,True)
            GPIO.output(24,False)
            time.sleep(5)
            GPIO.output(23,False)
        else:
            GPIO.output(23,False)
            print "Access Denied"
            GPIO.output(24,True)
            time.sleep(5)
            GPIO.output(24,False)
```

Referencias: https://behindthesciences.com/electronics/raspberry-pi-rfid-tag-reader/

Código final con las correcciones realizadas:

```
import time
import serial
from datetime import datetime
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)

#Tag1 = str('1800387799CE')
Tag1 = str('3F005F9330C3')
Tag2 = str('3F00ECF2D7F6')
GPIO.setup(23,GPIO.OUT)
GPIO.setup(24,GPIO.OUT)
GPIO.output(23,False)
GPIO.output(24,False)
PortRF = serial.Serial('/dev/ttyAMA0',9600)
while True:
   ID = ""
   read_byte = PortRF.read().decode("utf-8")
   if read_byte=="\x02":
      for Counter in range(12):
         read_byte=PortRF.read()
         ID = ID + str(read_byte)
         #print(hex(ord(read_byte)))
      ID=ID.replace("b'","").replace("'","")
      print(ID)
      now = datetime.now()
      if ID == Tag1:
         print("matched", now)
         GPIO.output(23,True)
         GPIO.output(24,False)
         #time.sleep(5)
         GPIO.output(23,False)
      else:
         if ID == Tag2:
            print("matched", now)
            GPIO.output(23,True)
            GPIO.output(24,False)
            #time.sleep(5)
            GPIO.output(23,False)
         else:
            GPIO.output(23,False)
            print("Access Denied")
            GPIO.output(24,True)
            #time.sleep(5)
            GPIO.output(24,False)
```