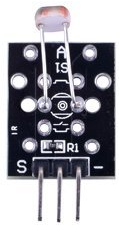
Photoresistor



Overview

Photoresistors (also called photo-conductive cells, photocells, CdS cells, or light-dependent resistors) are variable resistors that exhibit *photoconductivity.* This means they increase their conductivity (or decrease their resistance to conduction) as incident light intensity increases. Photoresistors are mainly used in various photoelectric control systems, such as navigation lights, street lights, and other controls systems that take action when the lights go out (or come on). They can be used in manufacturing contexts (e.g. automatic protection device), position detection, automatic camera exposure controllers, smoke alarms, and other photosensitive contexts as well.

In this course, we will use the Raspberry Pi to drive an LED light based on the photoresistor’s signal.

Experimental Materials

Raspberry Pi x1

Breadboard x1

Photoresistor x1

ADC0832 x1

LED (3-pin) x1

Dupont jumper wires

Experimental Procedure

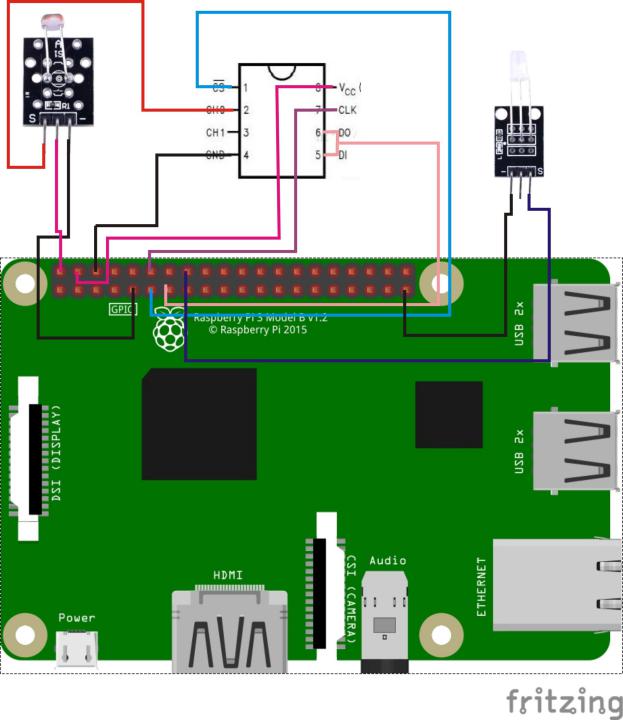
1. If you have not done so already, prepare your development system by installing the Python interpreter, RPi.GIO library, and wiringPi library as described in READ\_ME.TXT.
2. Install the ADC0832 analog/digital converter IC, photoresistor, and three-pin LED on your breadboard, and use Dupont jumper wires to connect them to each other and your Raspberry Pi as illustrated in the Wiring Diagram below. (The three-pin LED provided in this kit includes onboard series resistors, so no additional resistors are needed.)
3. Execute the sample stored in this experiment’s subfolder.

If using C, compile and execute the C code:  
cd Code/C  
gcc photoResistor.c -o photoResistor.out –lwiringPi  
./ photoResistor.out

If using Python, launch the Python script:  
cd Code/Python  
python photoResistor.py

1. Make experimental observations as you vary the ambient light falling on the photoresistor. As it becomes darker, the LED illuminates. As it becomes brighter, the LED extinguishes.

Wiring Diagram



AD0382 Pin position:

CS ↔ Raspberry Pi pin 11

CLK ↔ Raspberry Pi pin 12

DI ↔ Raspberry Pi pin 13

DO ↔ Raspberry Pi pin 13

CH0 ↔ Photoresistor pin "S"

VCC ↔ Raspberry Pi +5V

GND ↔ Raspberry Pi GND

Photoresistor pin position:

"S" ↔ ADC0382 pin CH0

"+" ↔ Raspberry Pi +5V

"-" ↔ Raspberry Pi GND

LED pin position:

"S” ↔ Raspberry Pi pin 16

"-" ↔ Raspberry Pi GND

Sample Code

Python code

#!/usr/bin/env python

import RPi.GPIO as GPIO

import ADC0832

import time

LedPin = 16

threshold = 120

def init():

ADC0832.setup()

GPIO.setup(LedPin, GPIO.OUT)

def loop():

while True:

res = ADC0832.getResult(0)

print 'res = %d' % res

if(res > threshold):

print 'It is night, light on...'

GPIO.output(LedPin, GPIO.HIGH)

else:

print 'It is already dawn, light off'

GPIO.output(LedPin, GPIO.LOW)

time.sleep(0.2)

if \_\_name\_\_ == '\_\_main\_\_':

init()

try:

loop()

except KeyboardInterrupt:

ADC0832.destroy()

print 'The end !'

C Code

#include <wiringPi.h>

#include <stdio.h>

typedef unsigned char uchar;

typedef unsigned int uint;

#define ADC\_CS 0

#define ADC\_CLK 1

#define ADC\_DIO 2

#define LedPin 4

#define threshold 120

uchar get\_ADC\_Result(uchar channel)

{

uchar i;

uchar dat1=0, dat2=0;

digitalWrite(ADC\_CS, 0);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,channel); delayMicroseconds(2);

digitalWrite(ADC\_CLK,1);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

for(i=0;i<8;i++)

{

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0); delayMicroseconds(2);

pinMode(ADC\_DIO, INPUT);

dat1=dat1<<1 | digitalRead(ADC\_DIO);

}

for(i=0;i<8;i++)

{

dat2 = dat2 | ((uchar)(digitalRead(ADC\_DIO))<<i);

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0); delayMicroseconds(2);

}

digitalWrite(ADC\_CS,1);

pinMode(ADC\_DIO, OUTPUT);

return(dat1==dat2) ? dat1 : 0;

}

int main(void)

{

uchar analogVal;

if(wiringPiSetup() == -1)

{

printf("setup wiringPi failed !");

return -1;

}

pinMode(ADC\_CS, OUTPUT);

pinMode(ADC\_CLK, OUTPUT);

pinMode(ADC\_DIO, OUTPUT);

pinMode(LedPin, OUTPUT);

while(1)

{

analogVal = get\_ADC\_Result(0);

printf("analogVal is %d.\n", analogVal);

if(analogVal > threshold)

{

printf("It is night,light on!\n");

digitalWrite(LedPin, HIGH);

}

else

{

printf("It is already dawn, light off!\n");

digitalWrite(LedPin, LOW);

}

delay(200);

}

return 0;

}

Technical Background

◆Maximum voltage: 150V DC

◆ Max Power: 100 mW

◆Ambient Temperature: -30°C … +70°C

◆ Spectrum Peak: 540nm

◆Response Time: rise: 20ms; drops: 30ms