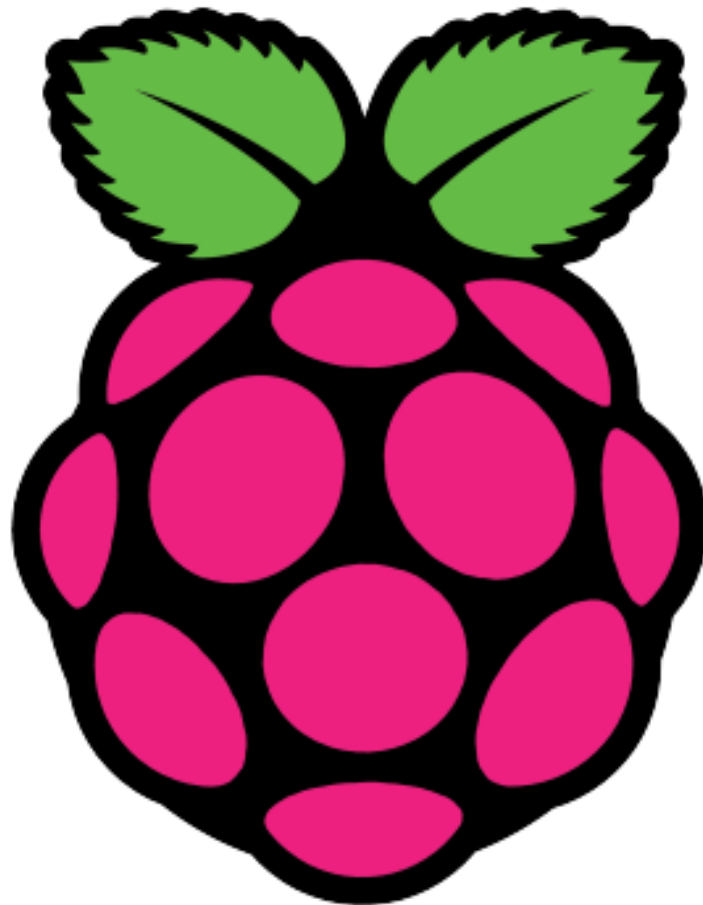


“Raspberry pi course”

ENG: AHMED MUBARAK

01032414034



SESSION NO.“10”

- KEYBOARD CONTROL
- L298N MOTOR DRIVER
- ROBOT CONTROLLED USING KEAYBOARD
- MCP3008 ADC
- POTINTIOMETER
- GAS SENSOR

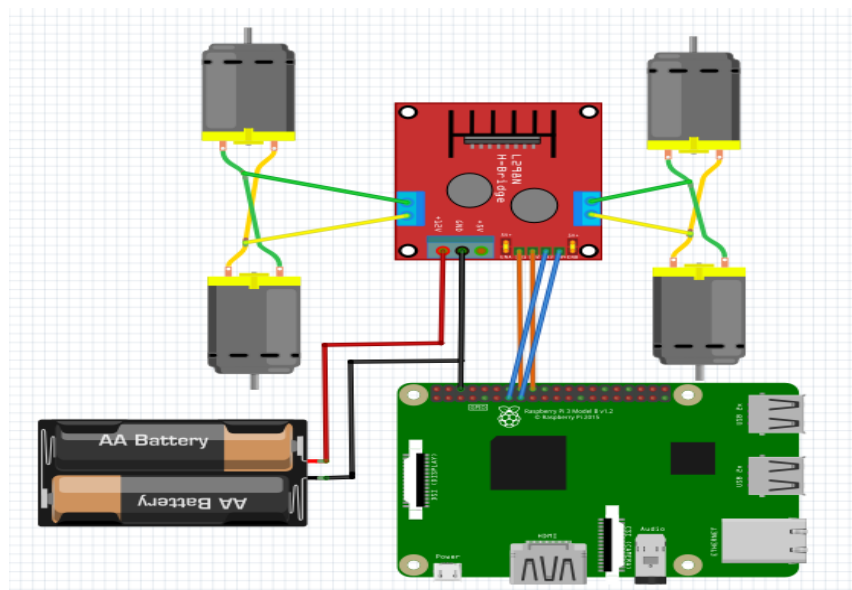
ENG.AHMED MUBARAK

01032414034

KEYBOARD CONTROL

```
# import curses
import curses
# Get the curses window, turn off echoing of keyboard to screen, turn on
# instant (no waiting) key response, and use special values for cursor keys
screen = curses.initscr()
curses.noecho()
curses.cbreak()
screen.keypad(True)
try:
    while True:
        char = screen.getch()
        if char == ord('q'):
            break
        elif char == curses.KEY_UP:
            print "up"
        elif char == curses.KEY_DOWN:
            print "down"
        elif char == curses.KEY_RIGHT:
            print "right"
        elif char == curses.KEY_LEFT:
            print "left"
        elif char == 10:
            print "stop"
finally:
    #Close down curses properly, inc turn echo back on!
    curses.nocbreak(); screen.keypad(0); curses.echo()
    curses.endwin()
```

L298N MOTOR DRIVER



ROBOT CONTROL USING KEAYBOARD

ROBOT CONTROL USING KEAYBOARD

```
# import curses, GPIO and time
import curses
import RPi.GPIO as GPIO
import time

#set GPIO numbering mode and define output pins
GPIO.setmode(GPIO.BOARD)
GPIO.setup(16,GPIO.OUT)
GPIO.setup(18,GPIO.OUT)
GPIO.setup(13,GPIO.OUT)
GPIO.setup(15,GPIO.OUT)

# Get the curses window, turn off echoing of keyboard to screen, turn on
# instant (no waiting) key response, and use special values for cursor keys
screen = curses.initscr()
curses.noecho()
curses.cbreak()
screen.keypad(True)
try:
    while True:
        char = screen.getch()
        if char == ord('q'):
            break
        elif char == curses.KEY_UP:
            GPIO.output(16,False)
            GPIO.output(18,True)
            GPIO.output(13,False)
            GPIO.output(15,True)
        elif char == curses.KEY_DOWN:
            GPIO.output(16,True)
            GPIO.output(18,False)
            GPIO.output(13,True)
            GPIO.output(15,False)
        elif char == curses.KEY_RIGHT:
            GPIO.output(16,False)
            GPIO.output(18,True)
            GPIO.output(13,True)
            GPIO.output(15,False)
        elif char == curses.KEY_LEFT:
            GPIO.output(16,True)
            GPIO.output(18,False)
            GPIO.output(13,False)
            GPIO.output(15,True)
        elif char == ord('d'):
            GPIO.output(18,True)
            GPIO.output(15,True)
            time.sleep(.5)
            GPIO.output(16,True)
```

```

GPIO.output(18,False)
GPIO.output(13,True)
GPIO.output(15,False)
time.sleep(.5)
GPIO.output(16,True)
GPIO.output(18,False)
GPIO.output(13,False)
GPIO.output(15,True)
time.sleep(.5)
GPIO.output(16,False)
GPIO.output(18,True)
GPIO.output(13,True)
GPIO.output(15,False)
time.sleep(.5)
GPIO.output(18,False)
GPIO.output(13,False)
elif char == ord('s'):
    GPIO.output(16,False)
    GPIO.output(18,False)
    GPIO.output(13,False)
    GPIO.output(15,False)

```

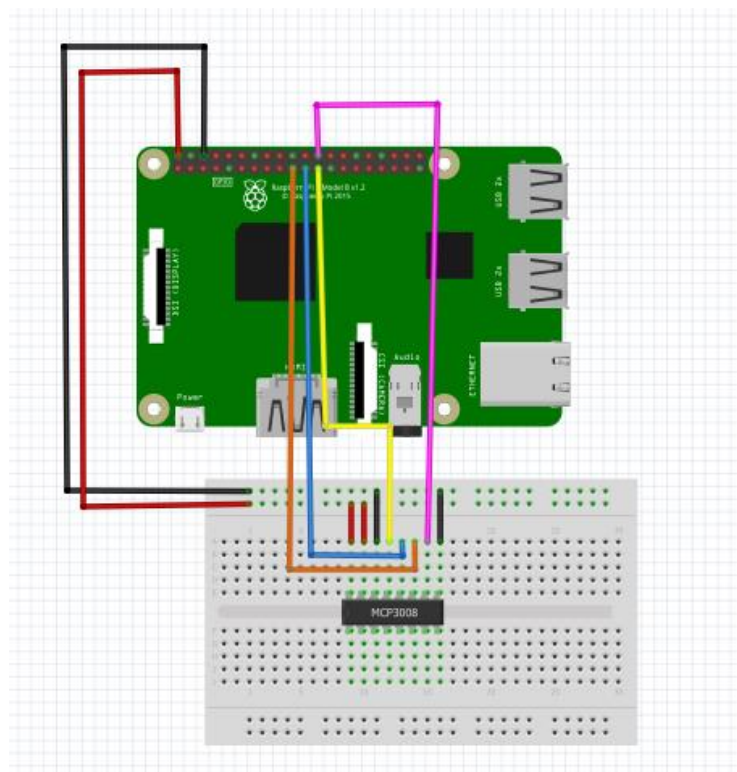
finally:

```

#Close down curses properly, inc turn echo back on!
curses.nocbreak(); screen.keypad(0); curses.echo()
curses.endwin()
GPIO.cleanup()

```

MCP3008 ADC



HOW TO INTERFACE MCP3008 ON RASPBERRY PI

1. Enable SPI

"sudo raspi-config"

select interfacing options

select SPI

2. Check to see if SPI is activated

"lsmod" -> should see something like spi_2835

3. Download Adafruit libraries

- git clone https://github.com/adafruit/Adafruit_Python_MCP3008.git
- cd Adafruit_Python_MCP3008
- sudo python setup.py install

EXAMPLE CODE

```
import time

# Import SPI library (for hardware SPI) and MCP3008 library.
import Adafruit_GPIO.SPI as SPI
import Adafruit_MCP3008

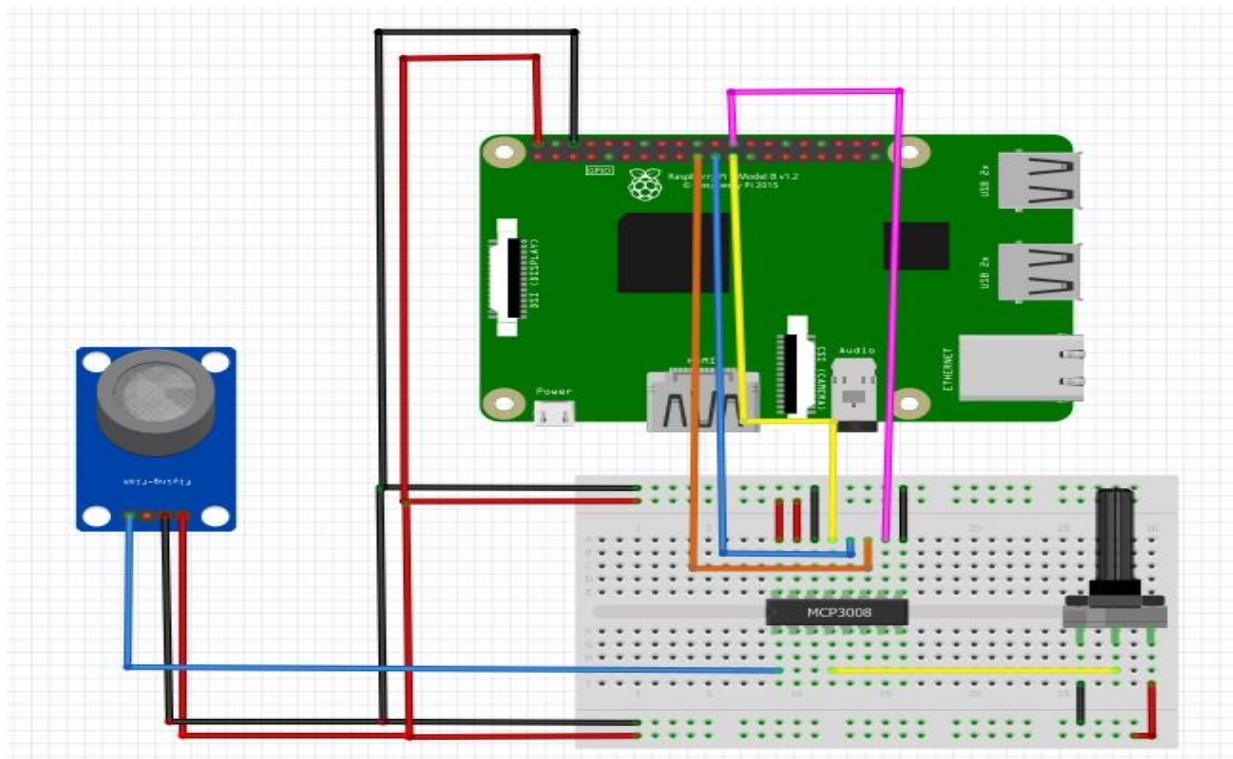
# Software SPI configuration:
CLK = 11
MISO = 9
MOSI = 10
CS = 8
mcp = Adafruit_MCP3008.MCP3008(clk=CLK, cs=CS, miso=MISO, mosi=MOSI)
# Hardware SPI configuration:
# SPI_PORT = 0
# SPI_DEVICE = 0
# mcp = Adafruit_MCP3008.MCP3008(spi=SPI.SpiDev(SPI_PORT, SPI_DEVICE))
print('Reading MCP3008 values, press Ctrl-C to quit...')
# Print nice channel column headers.
print('| {0:>4} | {1:>4} | {2:>4} | {3:>4} | {4:>4} | {5:>4} | {6:>4} | {7:>4} |'.format(*range(8)))
```

```

print('-' * 57)
# Main program loop.
while True:
    # Read all the ADC channel values in a list.
    values = [0]*8
    for i in range(8):
        # The read_adc function will get the value of the specified channel (0-7).
        values[i] = mcp.read_adc(i)
        # if values[0] > 300 :
        #     print("good")
        # else :
        #     print("bad")
    # Print the ADC values.
    print('| {0:>4} | {1:>4} | {2:>4} | {3:>4} | {4:>4} | {5:>4} | {6:>4} | {7:>4} |'.format(
*values))
    # Pause for half a second.
    time.sleep(0.5)

```

POTENTIOMETER AND GAS SENSOR



***FINALLY: I HOPE I HAVE
SUCCEEDED IN SAMPLIFYING THIS
CONTENT***

SEE YOU SOON IN ANOTHER COURSE

With my best wishes:

ENG : AHMED MUBARAK
