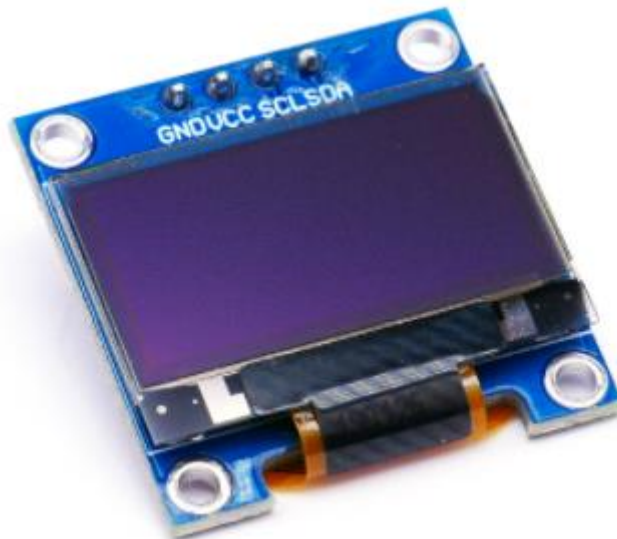


Podstawy Techniki Mikroprocesorowej

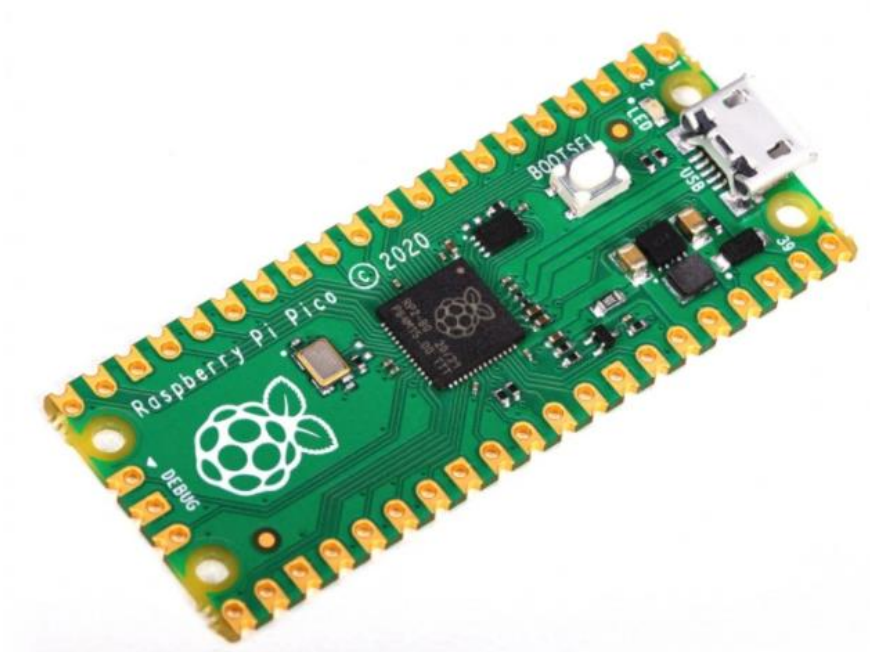
Opis Projektu

Czujniki ultradźwiękowe wykorzystują impulsy dźwiękowe i proste obliczenia, aby określić odległość między sobą a obiektami znajdującymi się przed nimi. Są często używane w robotach, aby upewnić się, że robot nie przejdzie lub nie wpadnie na przeszkodę. W tym projekcie wykorzystaliśmy ultradźwiękowy czujnik odległości HC-SR04, aby określić odległość obiektu od naszego Raspberry Pi Pico. Ta kompilacja wykorzystuje wyłącznie czujnik ultradźwiękowy zasilany 5V, używanego w GPIO Raspberry Pi Pico. HC-SR04P i HC-SR04+ są kompatybilne 3V i 5V zasilaniem, co czyni je idealnymi do projektów Raspberry Pi Pico, Pi i Arduino.

Spis komponentów



Rysunek 1: Wyświetlacz OLED ssd1306



Rysunek 2: Raspberry Pi Pico

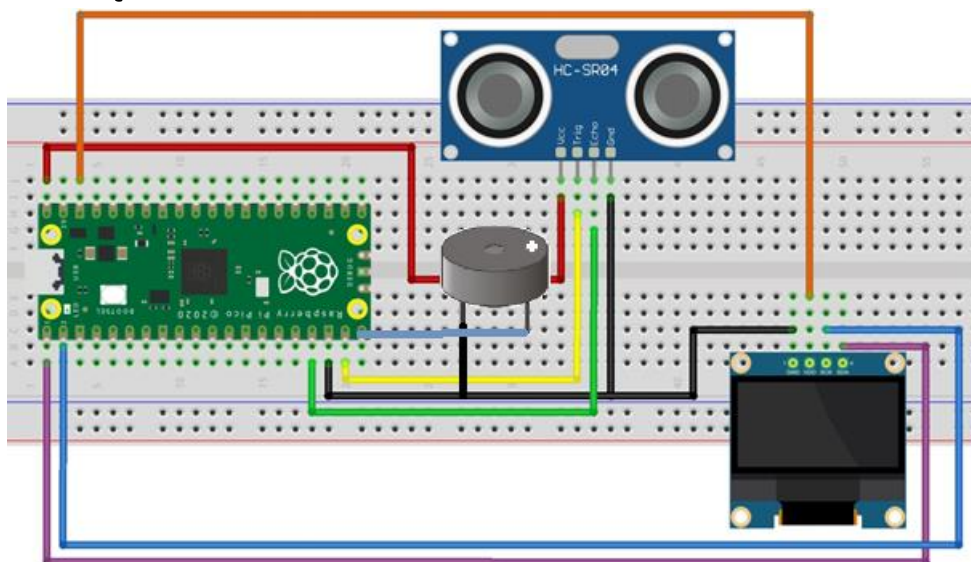


Rysunek 3: Ultra soniczny czujnik odległości HC-SR04



Rysunek 4: Aktywny Buzzer 5V

Schemat Połączeń



Rysunek 5: Schemat połączeń

Kod w Pythonie

```
1 #import bibliotek
2 # biblioteka machine służy do kotrolowania pinów GPIO
3 from machine import Pin,PWM
4 import utime
5 from machine import Pin, I2C
6 from ssd1306 import SSD1306_I2C
7 import framebuf #biblioteka potrzebna do przesłania danych na wyświetlacz
8 import math
9 import utime
10 # Rozdzielczość wyświetlacza
11 WIDTH = 128
12 HEIGHT = 64
13
14 speaker = PWM(Pin(15)) #deklaracja input buzzera
15 sda=machine.Pin(0) #deklaracja lini danych
16 scl=machine.Pin(1) #deklaracja lini zegara
17 i2c=machine.I2C(0,sda=sda, scl=scl, freq=400000)#deklaracja magistrali I2C
18 oled = SSD1306_I2C(128, 64, i2c)
19 oled.text("STechiezDIY !!!",5,5)
20 oled.text("Pico",5,15)
21 oled.text("HCSR04",5,25)
22 oled.text("SSD1306",5,35)
23 trigger = Pin(3, Pin.OUT) #deklaracja impulsu wyzwolenia. W naszym przypadku do wysłania fali
24 echo = Pin(2, Pin.IN) #deklaracja pinu echo. Służy on do odebrania wracającego sygnału
25 def get_distance():
26     trigger.low()#ustawienie trigger pinu na stan niski
27     utime.sleep_us(2)#zapauzowanie na 2s
28     trigger.high()#ustawienie pinu trigger na stan wysoki na 5s i zatrzymanie go
29     utime.sleep_us(5)
30     trigger.low()
31     #Jeśli nie zostanie odebrany żaden impuls echa,
32     #zaktualizuj zmienną, signaloff, tak aby zawierała czas w ms
33     while echo.value() == 0:
34         signaloff = utime.ticks_us()
35
36 #Kolejna pętla while,sprawdza czy echo zostało odebrane.
37 while echo.value() == 1:
38     signalon = utime.ticks_us()
39     timepassed = signalon - signaloff # zmienna przechowywująca czas w jakim sygnał wyszedł, odbił się i wrócił
40     distance = (timepassed * 0.0343) / 2 #obliczanie odległości na podstawie czasu i prędkości dźwięku
41     print("The distance from object is ",distance,"cm")
42     return distance
43 while True:
44     #implementacja wyświetlania odległości na wyświetlaczu
45     oled.fill(0)
46     ret_val = get_distance()
47     oled.text("Distance:",0,0)
48     oled.text(str(ret_val) + " cm",0,10)
49     oled.show()
50     utime.sleep(1)
51     #implementacja wykrywania odległości mniejszej niż 20cm i działanie buzzera
52     if get_distance() < 2000:
53         speaker.duty_u16(3000)
54         speaker.freq(1700)
55         utime.sleep(0.05)
56         speaker.duty_u16(0)
57         utime.sleep(get_distance() / 1000)
```

Opis działania programu

W module ultradźwiękowym HCSR04 podajemy impuls wyzwalający na trigger PIN, tak aby wygenerował on ultradźwięki o częstotliwości 40 kHz. Po wygenerowaniu ultradźwięków tj. 8 impulsów o częstotliwości 40 kHz stan Echo zmienia się na wysoki. Stan ten nie zmienia się dopóki sygnał nie wróci z powrotem. Tak więc szerokość Echa będzie czasem, w którym dźwięk dotrze do obiektu i powróci z powrotem. Gdy otrzymamy czas, możemy obliczyć odległość, ponieważ znamy prędkość dźwięku.

Czujnik HC-SR04 może mierzyć do 2 cm - 400 cm. Moduł ultradźwiękowy wygeneruje fale

ultradźwiękowe, które są powyżej progu słyszalności człowieka. W naszym przypadku będziemy nadawać na częstotliwości 40Khz. W naszym projekcie użyliśmy jeszcze brzęczek aby sygnalizował dźwiękiem jeśli przedmiot znajduje się w odległości mniejszej niż 20cm. HC-SR04, buzzer i SSD1306 posiada predefiniowane biblioteki co znacznie ułatwia stworzenie takiego projektu.

Wnioski

Komponenty które użyliśmy posiadają predefiniowane biblioteki które znacznie ułatwiają pracę. Nasz czujnik oraz magistrala I2C nie są najszybsze więc projekt ten nie znalazł by zastosowania gdzie odległość musi być przetwarzana w czasie rzeczywistym.