"Helping Hand"

Cel projektu

Wykonanie urządzenia mającego na celu pomoc osobom niewidomym w samodzielnym poruszaniu się Projekt i funkcje urządzenia

Urządzenie noszone w dłoni

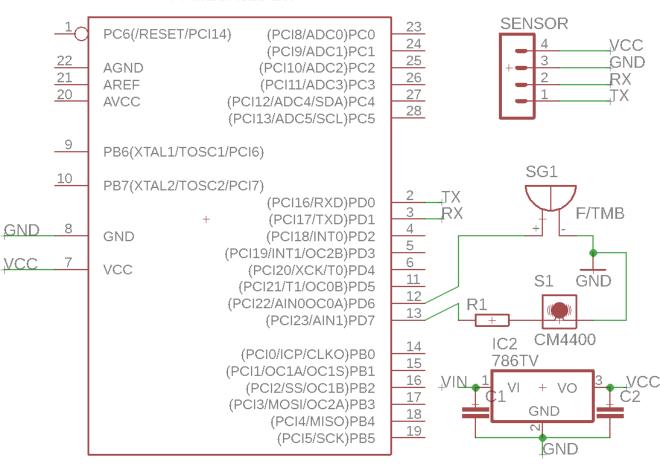
Sygnały dźwiękowe

Wibracje

Zasilanie bateryjne, bezprzewodowość

ATMEGA328-20P



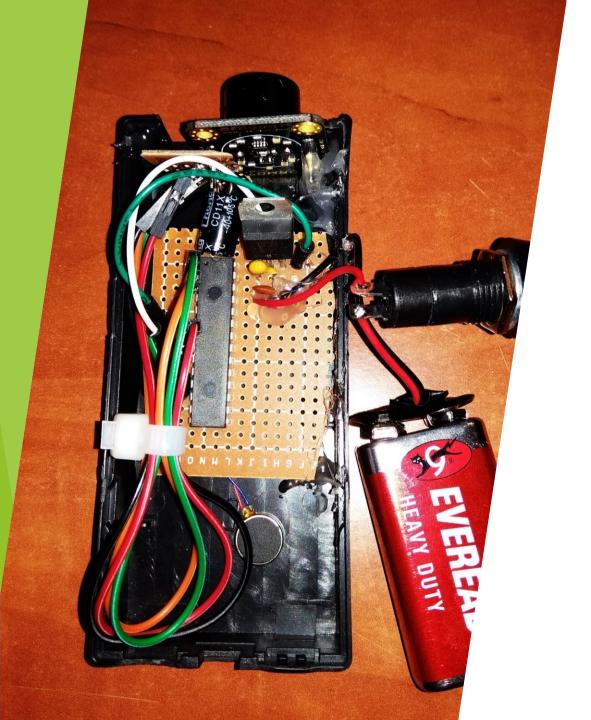


Kod C++

```
#include <avr/io.h> // input/output library
 #define F CPU 8000000UL //CPU clock define
 #include <avr/interrupt.h> // interrupts library
 #include <util/delay.h> //delay library
 #define USART_BAUDRATE 19200 //usart communication baudrate
 #define BAUD PRESCALER (((( F CPU / 16) + ( USART BAUDRATE / 2) ) / ( USART BAUDRATE )) - 1) //
 #define header H 0x55 //Header
 #define device Addr 0x11 //Address
 #define data Length 0x00 //Data length
 #define get Dis CMD 0x02 //Command: Read distance
                 (header_H+header_L+device_Addr+data_Length+get_Dis_CMD); //checksum is a sequ
 int checksum =
 unsigned char sensor_data[8]; // 8-elements matrix for sensor data
 unsigned char CMD[5]={header_H,header_L,device_Addr,data_Length,get_Dis_CMD}; //distance comman
     #define BUFFER_SIZE 8  // size of Rx ring buffer.
     volatile int rxn=0; // buffer 'element' counter.
     volatile char rx[BUFFER SIZE]; // buffer of 'char'.
     volatile uint8_t rxFlag = 0; //rxFlag
void UART send( unsigned char data)
     while ((UCSR0A & (1<<UDRE0)) == 0) {} //if UDRE0 = 1 (to jest bit w rejestrze UCSR0A) to wt
     UDR0 = data; // UDR0 - UDRn - UART I/O Data Register (buffer danych do wyslania oraz odebr
```

```
for(int i=0;i<5;i++) //wyslanie po kolei wszystkich bajtow do czujnika (checksum osobno bo nie jest w tablicy)
     UART_send(CMD[i]);
 UART_send(checksum);
 _delay_ms(20);
if (rxFlag == 1) // process received data in buffer.
        // The following loop simply returns the current contents of the entire buffer, each time the rxFlag goes high.
    for (int i=0;i<rxn;i++) //rxn - ilosc elementow w wirtualnym bufferze</pre>
        if (i<8)
            sensor_data[i] = rx[i]; //data from v buffer to sensor data array
    rxFlag=0; //reset the rxFlag so it waits for another interrupt to set it to 1, signifying new data
    int distance =((sensor_data[5]<<8)|sensor_data[6]); //calculate the distance (2 bytes - High and Low) //distance in [cm]
```

```
if (distance>70) //distance ok
       PORTB = 0b00000010;
       PORTD = 0b00000000;
   if (distance<70 && distance>0) //distance too close
       PORTD = 0b110000000;
                                  //vibrate and buzzer on
       delay(2^(distance*3));
       PORTD = 0b100000000;
                                  // buzzer off
       delay(2^(distance*3));
ISR (USART_RX_vect) //interrupt handler
    while ( !(UCSR0A & (1<<RXCO)) ){} /* Wait for data to be received */
    if (rxn==BUFFER_SIZE)// if BUFFER_SIZE is reached, reset to start of buffer.
            rxn=0;
    if(rxn == 7)
        rxFlag=1; // notify main of receipt of data.
    rx[rxn++] = UDR0; // increment rxn and return new value. Transfering data from avr buffer (UDR0) to virtual buffer (rx[])
```



Zdjęcie urządzenia

Podsumowanie

Udało się skonstruować urządzenie wykrywające przeszkody

Urządzenie wygodne w użytkowaniu

Zbyt mała ilość czujników

Wymagane dopracowanie oprogramowania