

سنسورهای اولتراسونیک

با توجه به اینکه محدوده فرکانسی شنوایی انسان بین 20 هرتز تا 20 کیلو هرتز میباشد و اکثر اصوات موجود در طبیعت به این محدوده اختصاص دارند لذا از این محدودیت می توان در زمینه های گوناگون به صورت مفید استفاده کرد.

استفاده از این سنسور ها موجب میشود علاوه بر اینکه اصوات تولیدی گوش انسان را آزار ندهد ، اکثر صداهای اطراف ما تداخلی در عملکرد سنسور نداشته و اطلاعات تا حد زیادی حفظ شود.

ماژولهای اولترا سونیک خانواده SRF

ماژولهای اولتراسونیک خانواده SRF به عنوان مثال SRF02، SRF05 و SRF08 امروزه در صنایع مختلف می توانند به کار گیری شوند و از جمله پر کاربرد ترین محل استفاده این ماژولها می توان به صنعت روبات اشاره کرد.

بسته به نوع و شماره ماژولها دقت ، نوع ارتباط با میکرو یا PC و برد های مختلفی را از آنها می توان انتظار داشت . در این مقاله سعی شده است . به عنوان مثال SRF02 با برد بین 16Cm تا 6 متر دارای دقت اندازه گیری 1Cm یا 1 اینچ یا یک میکرو ثانیه باشد.

در این مقاله سعی شده است به طرز راه اندازی مدل SRF02 پرداخته شود.

SRF02

در این ماژول از یک سنسور Transreceiver برای عمل ارسال و دریافت امواج فراصوت استفاده شده است . این ماژول را میتوان به صورت ارتباط سریال UART و یا I2C با سخت افزار خود ارتباط داد.



شکل شماره (1)

برای ارتباط با این ماژولها باید دو بیت را به ترتیب به ماژول (یا ماژولها) ارسال کنیم که اولین بیت حاوی کد یا نام ماژول از 0 تا 15 بوده و برای انتخاب ماژولی که میخواهیم با آن ارتباط برقرار کنیم به کار می رود. بیت دوم کد مربوطه تعیین نوع عملکرد ماژول می باشد که در جدول شماره 1 آمده است.

در جدول شماره (1) مشخص است که برای دریافت اطلاعات از ماژول برحسب اینچ عدد 80 و بر حسب سانتی متر باید 81 و بر حسب میکرو ثانیه باید عدد 82 را به میکرو در بیت دوم ارسال کنیم. کدهای 83 تا 85 معادل همان کدهای 80 تا 82 هستند با این تفاوت که اگر از کدهای 83 تا 85 استفاده کنیم، بلافاصله پس از آماده شدن اطلاعات در ماژول خودبخود به میکرو یا PC ارسال میشود ولی در صورت استفاده از کدهای 80 تا 82 باید بعد از اینکه 65 میلی ثانیه برای آماده شدن اطلاعات در ماژول صبر کردیم، باید کد 94 را ارسال کنیم تا ماژول شروع به ارسال اطلاعات کند. کدهای 86 تا 91 نیز همان معادل کدهای 80 تا 85 می باشند با این تفاوت که در دسته 80-85 بین کدهای ارسالی فاصله خودبخودی ایجاد می شود ولی در دسته 86-91 این فاصله را باید توسط کنترلگر (میکرو یا PC) ایجاد کنیم.

Command		Action
Decimal	Hex	
80	0x50	Real Ranging Mode - Result in inches
81	0x51	Real Ranging Mode - Result in centimeters
82	0x52	Real Ranging Mode - Result in micro-seconds
83	0x53	Real Ranging Mode - Result in inches, automatically Tx range back to controller as soon as ranging is complete.
84	0x54	Real Ranging Mode - Result in centimeters, automatically Tx range back to controller as soon as ranging is complete.
85	0x55	Real Ranging Mode - Result in micro-seconds, automatically Tx range back to controller as soon as ranging is complete.
86	0x56	Fake Ranging Mode - Result in inches
87	0x57	Fake Ranging Mode - Result in centimeters
88	0x58	Fake Ranging Mode - Result in micro-seconds
89	0x59	Fake Ranging Mode - Result in inches, automatically Tx range back to controller as soon as ranging is complete.
90	0x5A	Fake Ranging Mode - Result in centimeters, automatically Tx range back to controller as soon as ranging is complete.
91	0x5B	Fake Ranging Mode - Result in micro-seconds, automatically Tx range back to controller as soon as ranging is complete.
92	0x5C	Transmit an 8 cycle 40khz burst - no ranging takes place
93	0x5D	Get software version - sends a single byte back to the controller
94	0x5E	Get Range, returns two bytes (high byte first) from the most recent ranging.
95	0x5F	Get Minimum, returns two bytes (high byte first) of the closest range measurable - see Autotune section
96	0x60	Force Autotune Restart - same as power-up. You can ignore this command.
160	0xA0	1st in sequence to change I2C address
165	0xA5	3rd in sequence to change I2C address
170	0xAA	2nd in sequence to change I2C address

جدول شماره (1)

پس از ارسال کدهای اشاره شده ماژول اطلاعات را در قالب دو بایت پشت سر هم ارائه می کند که با کنار هم قرار دادن این بایتها (بصورت اولین بایت با ارزشترین آنها) اطلاعات بر حسب میکرو ثانیه ، اینچ یا سانتی متر بصورت 16 بیتی بدست می آید.
بر نامه زیر به عنوان نمونه می تواند اطلاعات را با پیکره بندی ماژول شماره 0 دریافت کند:

```
Enable interrupts  
$baud = 9600
```

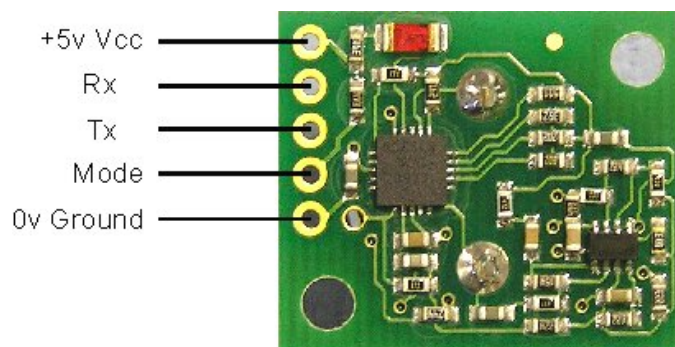
```
dim a as byte  
dim b as byte  
dim c as word
```

```
printbin 0  
waitms 50  
printbin 84
```

```
a= waitkey()  
b= waitkey()
```

```
c=a*256  
c=c+b
```

البته برنامه فوق که به زبان بیسیک در کامپایلر بیسکام معتبر است ، بصورت سریال UART با ماژول ارتباط برقرار میکند. باید در نظر داشت که در صورت استفاده از این ماژول بصورت UART باید پایه MODE آن را به گراند یا همان 0 منطقی متصل کرد (پایه ها مطابق شکل 2).



شکل شماره (2)

تغییر دادن آدرس یا نام ماژول

16 عدد از این ماژولها را می توان بصورت همزمان و بصورت موازی باهم به یک میکرو یا کامپیوتر وصل کرد بطوری که هر کدام را از 0 تا 15 باید نامگذاری کرد. کد یا نام ماژولها بصورت پیشفرض 0 می باشد ما با استفاده از کدهای اختصاص داده شده توسط کمپانی سازنده که در جدول شماره 1 آورده شده اند، می توانیم کد یا نام هر ماژول را تغییر دهیم. به عنوان مثال برای تغییر نام یک ماژول از 0 به 5 باید کدهای زیر را به ترتیب به ماژول ارسال کنیم :

```
printbin 0
printbin 160
printbin 0
printbin 170
printbin 0
printbin 165
printbin 0
printbin 5
```

پس از آدرس دهی در صورتی که آدرس ذخیره شده در ماژول توسط کاربر فراموش شود میتوان با برقراری تغذیه ماژول و بدون ارسال کد با فلش های LED ماژول به کد آن پی برد. بطوری که پس از برقراری تغذیه یک فلش ممتد نسبتاً طولانی و پس از آن به شماره آدرس ماژول فلشهای کوتاه و پشت سر هم ایجاد میشود(مطابق اطلاعات جدول شماره 2).

ماژول در حین کار (ارسال کدها از طرف میکرو و ارسال اطلاعات از طرف ماژول) دائماً فلشهای بسیار کوتاهی خواهد زد.

برنامه زیر نیز یک نمونه برنامه به زبان C برای اتصال ماژول از درگاه I2C می باشد :

```
SRF02 srf02[3] =
{
  SRF02(0x70, SRF02_CENTIMETERS),
  SRF02(0x71, SRF02_CENTIMETERS),
  SRF02(0x72, SRF02_CENTIMETERS)
};

unsigned long nextPrint = 0;

void setup()
{
  Serial.begin(9600);
```

```

SRF02::setInterval(500);
}

void loop()
{
  while (true)
  {
    SRF02::update();
    if (millis() > nextPrint)
    {
      Serial.print(srf02[0].read());
      Serial.print(",");
      Serial.print(srf02[1].read());
      Serial.print(",");
      Serial.print(srf02[2].read());
      Serial.println();
      nextPrint = millis () + 1000;
    }
  }
}

```

Address		Long	Short
Decimal	Hex	Flash	flashes
0	00	1	0
1	01	1	1
2	02	1	2
3	03	1	3
4	04	1	4
5	05	1	5
6	06	1	6
7	07	1	7
8	08	1	8
9	09	1	9
10	0A	1	10
11	0B	1	11
12	0C	1	12
13	0D	1	13
14	0E	1	14
15	0F	1	15

جدول شماره (2)

