- Home
- News & Blog
  - News and notes
  - Observer's calendar
  - Bookmarks
    - Fun
    - Astronomical software
    - System tools
    - Astronomy sites
- Articles
  - Programming
  - Windows
  - Astronomy
  - Electronics (& Arduino)
- Products
  - Last Update Module (Joomla!)
  - jQuery++ Integrator
  - Parental Time Control
  - Simple Account Timer
  - Joomla! plugins
  - Free Templates
- Contacts
  - Send e-mail
  - Follow me on Twitter

# tushev.org

## Semyon Tushev's web site

## Measuring frequency with Arduino

Thursday, 03 March 2011 22:59 | Written by tushev | |

There are many signals that contain necessary information in frequency. For instance, SparkFun's HH100D humidity sensor outputs measurements via frequency signal *(although it has $i^2$C interface, those pins are used for reading calibration values. Actual measurements com via FOUT line)*.

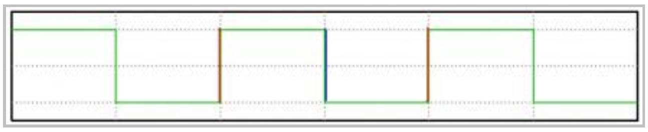
There are two ways to measure a frequency with Arduino:

- Using FreqCounter library . It can measure frequencies up to several MHz, and is very precise. However, it is fixed to digital pin 5. Also it may affect PWM outputs, increasing their duty cycles. In case you don't use PWM and you need to measure no more than one signal, FreqCounter is an excellent choice
- Using pulseIn() function . The technique I'm suggesting below is free from the limitations of FreqCounter, sou you can use it on any pin. However, it's minimal wave period is 10 uS, which stands for max frequency of 100 kHz

Here is a documentation for pulseIn function.

Most common mistake that people do when measuring frequency via pulseIn is getting one measurement. In many cases, frequency is not very stable, so we need to calculate an average value over a time. Something like 1024 samples will be enough, however, for better results, you may increase the value - up to 32767 samples (you can, certainly, choose any value, but, (for HH100D), there's almost no difference in measured signal frequency for 4096 and 32767 samples. The last value, however, takes up to 5 seconds to measure - depending on frequency, of course.)

We will suggest that we have a square wave with 50% duty cycle, and measure the length of the pulse. Then we'll multiply that value for 2 to obtain signal period: *(the distanse between red lines on the image)*:



```
1. long getFrequency(int pin) {
2.   #define SAMPLES 4096
3.   long freq = 0;
4.   for(unsigned int j=0; j<SAMPLES; j++) freq+= 500000/pulseIn(pin, HIGH, 250000);
5.   return freq / SAMPLES;
6. }
```

We wait for the pulse to occur in 250 mS. If you are using very slow signals, increase this value as you need. (line 4)

I use value of 500 000 instead of 1 000 000 uS, because I want to get signal's period, not pulse width (remember, we have duty cycle of 50%, so T = $2*t_{pulse}$).

Do not forget to configure your pin as input!

## Comments

### # RE: Measuring frequency with Arduino — Tom 2011-03-31 07:58

0

Thanks for the article. I am currently trying to make a HR monitor that will output to a set of LED's.

I have a good output signal from my sensor and it seems like this could accurately extract my frequency (~0.3 - 4 Hz).

Would it be easy to output this to a row a 5-6 LED's?

Thanks,
Tom

Reply | Reply with quote | Quote | Report to administrator

### # RE: RE: Measuring frequency with Arduino — tushev 2011-03-31 12:24

0

I think it will. If you need linear output, Arduino language has a special function for this - http://arduino.cc/en/Reference/Map. Unfortunately, it accepts only integer numbers, but you can easily adapt its code (it is published on the page) by replacing **long** with **float**. If you need another mapping between levels (logarithmic, for instance), then you have to create your own func.

The simplest way of LED output is just attaching them to Arduino pins 6-12 directly:

Quote:

```
numLEDs = floatMap(freq, 0.3, 4, 0, 6); //floatMap is your function, described above.
for (int pin=6; pin<13; pin++) {
if(numLEDs>0)digitalWrite(pin, HIGH);
else digitalWite(pin, LOW);
numLEDs--;
}
```

(There are other ways, however, like using shift registers, i2c-controlled led bars etc...)

Another question is that FreqCounter library may not measure such low-frequency pulses correctly (it's only a suggestion, though, I did not work with such freq.range). In this case, you may want to use pulseIn, as described in the article.

Reply | Reply with quote | Quote | Report to administrator

### # RE: Measuring frequency with Arduino — Tom 2011-04-04 00:56

0

I've tried with FreqLibrary, but it is confusing me a lot. I think that pulseIn may work better for what I'm trying to do.

If I use pulseIn, does it matter whether I calculate the HIGH pulses or LOW pulses? Because calculating LOW would save me a bit of power.

Also, would it be easier for me to just calculate the amount of pulses that happen during a duration of time (say 10 seconds)? I am not trying to output this to a display, so I don't need an exact HR value, just ranges. (e.g. it would work if LED1 was 10-12 pulses, LED2 was 12-14 pulses, etc.) Thanks for your help

Reply | Reply with quote | Quote | Report to administrator

### # RE: RE: Measuring frequency with Arduino — tushev 2011-04-04 01:34

0

I don't think that high or low pulses save something.. In both cases, ATMEGA is waiting for pin value change, and counts the time. It should take almost the same amount of energy. The difference would be nanoJoules.
And the amount of energy the signal contains depends on its duty cycle (for square wave) - I mean, depends on generation process, not measurement...

However, HIGH or low values may play a role in getting stable results. When I used pulseIn techniques for my HH10D humidity sensor, I had to count timings for low-pulses (the sensor outputs a square wave with 50% duty cycle, with equal high and low periods - see the image for this article). For some reason, periods I got for high-pulses were fluctuating, while low periods were more stable. But this depends on your equipment. It's always better to measure frequency with third-party tool (oscilloscope, for instance), and only then try to use Arduino for measurements. And always try all variants!

As for FreqCounter, you need only 4 lines of code to get frequency from pin 5(not counting #include):

Quote:

```
//Get Frequency
FreqCounter::f_comp= 8; // Set compensation to 12
FreqCounter::start(1000); // Start counting with gatetime of 1000ms
while (FreqCounter::f_ready == 0) // wait until counter ready
freq=FreqCounter::f_freq; // read result freq must be of type long
```

See this article as an example: http://tushev.org/articles/electronics/46-interfacing-hh10d-with-arduino

As for counting pulses - maybe... It's just another ( a bit longer) way of doing the same - mapping. However, you should count those 10 seconds manually.

Reply | Reply with quote | Quote | Report to administrator

### # RE: Measuring frequency with Arduino — Tom 2011-04-04 01:46

0

Also, if you could help me out with syntax, I would really appreciate it. I'm brand new to Arduino and only have minimal C++ background.

Reply | Reply with quote | Quote | Report to administrator

### # RE: RE: Measuring frequency with Arduino — tushev 2011-04-04 01:50

What exactly?

0

Reply | Reply with quote | Quote | Report to administrator

### # RE: RE: RE: Measuring frequency with Arduino — Tom 2011-04-04 02:32

0

I honestly just realized that I had to integrate a microcontroller into the project last week, which I thought I could do completely analogously, so I have pretty much no idea what I'm doing any cannot find anyone around to really help me out.

I know basic set up and I've tried to teach myself a decent amount, but I just feel like the learning curve is a bit steep. I feel like I just need some good direction to get started, you know?

Reply | Reply with quote | Quote | Report to administrator

### # RE: RE: RE: RE: Measuring frequency with Arduino — tushev 2011-04-06 01:06

0

One of the best way of learning Arduino (and other uC) is practice. Just try provided examples, and try to figure out how they work. Move from easy ones to more complicated, and in some time you will realise that you actually understand a lot.

Another important thing is: try everything for yourself. Try even creating your own approaches.

Try to figure out, what is uC( in fact, a small computer with easy-to-use hardware interface), how does it communicate with external world.

Some experience in desktop programming would also be useful.

If you are asking about books, I learned everything from documentation and articles on the web. However, I've looked through (but not read) 'Getting started with Arduino' - http://my.safaribooksonline.com/book/hardware/arduino/9780596155704/, you can also google for it - i think it may help a bit.

Reply | Reply with quote | Quote | Report to administrator

### # RE: Measuring frequency with Arduino — Tom 2011-04-04 02:50

0

I've also seen instances where "attachInterrupt" is used, would that work any better?

Reply | Reply with quote | Quote | Report to administrator

### # RE: RE: Measuring frequency with Arduino — tushev 2011-04-06 01:08

0

It depends. Every realization does its work, and with its own peculiarities.

Take in mind, that inside interrupt function no millis are being counted (see Arduino site for more info)

Reply | Reply with quote | Quote | Report to administrator

### # RE: Measuring frequency with Arduino — Steve Gough 2011-04-17 08:26

0

Many thanks; I needed to write code to get freq data of

Reply | Reply with quote | Quote | Report to administrator

Refresh comments list
RSS feed for comments to this post

## Add comment

If you are asking for support, there's a special contact form for this.
However, if your question is useful to other people, **feel free to ask for support in the comments**. HTML is not allowed, use contact form for such cases.

Do not forget to specify your Joomla! version (if it's a question about a plugin). Also, if it is possible and/or necessary, **do not forget to provide a live URL related to the question**.
If you have a Gravatar it will be displayed here.

Name (required)

E-mail (required, but will not display)

Website

Title

2000 symbols left

□ Notify me of follow-up comments

Refresh

Send

JComments

## Quick Links

- Last Update Module
- Observer's calendar
- Free Templates
- Simple Account Timer
- FlowPlayer Reloaded
- jQuery++ Integrator
- jQuery for beginners
- Ask for support

## Recent publications

- Arduino and watchdog timer
- Modified EEPROM library for Arduino
- Interfacing HH10D with Arduino
- Compensating the difference between FreqCounter and pulseIn measurements
- Measuring frequency with Arduino
- How it works: DS18B20 and Arduino
- Free templates
- January 2011
- The Sun is smiling
- Preventing Arduino from auto-reset when COM port opens\closes

## Latest Tweets

Join the conversation

## News and updates

☑ jQuery++ Integrator

☑ FlowPlayer Reloaded

☑ All my extensions

Name

Email, never shared

Receive ○ Text ● HTML

Subscribe now

## Tag cloud

hh10d windows

products

astronomy parental

control solution

arduino tutorial

calendar c# jquery

programming

joomla stars last

update tips beginners

php jquery++ integrator

speech

Legal information | Privacy policy

Best viewed with Mozilla Firefox.