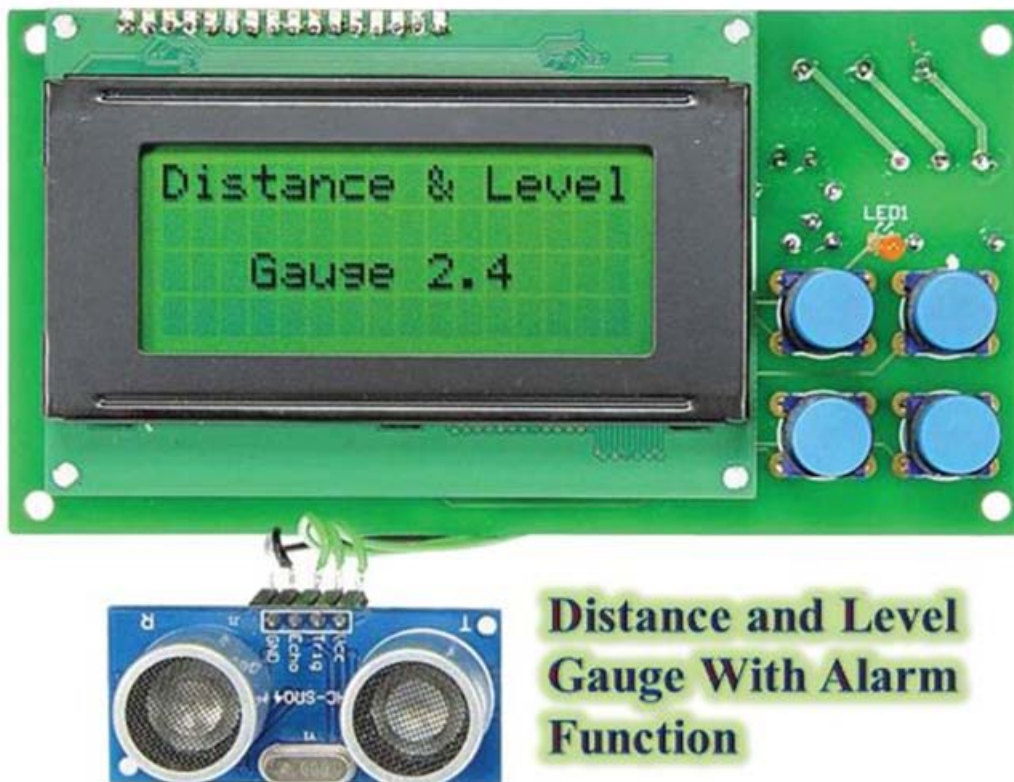


Ultrasonic Distance & Level Gauge Meter

Small, widely available and dirt-cheap ultrasonic sensors enable us to design both simple and lavish measuring devices for all kinds of ranging. Take one of these modules, add a display, a couple of press-buttons and a micro-controller loaded with software, and you now have all the ingredients for a circuit of this kind.



- Operating voltage: 5VDC.
- Temperature range: 0–70°C.
- Diffusion angle: 15°.
- Operating frequency: 40kHz.
- Trigger input signal: 10μs.
- Written in BASCOM-AVR BASIC language.
- Can be construct and program with Arduino boards.

Table of Contents

1. Introduction	3
2. Circuit and Components.....	4
Components List	6
3. The Control Firmware	6
4. Construction.....	7
5. Using Bascom-AVR with Arduino	9
6. Key Components	12

1. Introduction

Do you need to know how full your oil tank or water tank is, without requiring vast computational effort or expert knowledge? You fill a large barrel with water from the garden hose and would like to cut off the water flow using a magnetic valve when it has reached the desired level? Or maybe you would just like to measure the distance between two objects? The possibilities using the level and range meter described here are virtually limitless! The extent to which cylindrical or cuboid containers are full can be determined either mechanically with a float and potentiometer, or else capacitively with ultrasonics or a laser device. Both approaches, mechanical and capacitive, require auxiliary components such as floats and sensors. If accuracy of measurement is your prime consideration a laser is definitely the best solution, yet even this method has its pitfalls. For example, fog and steam can cause reflections for the laser, falsifying the result. Since influences of this kind affect ultrasonics hardly at all, I settled on ultrasonic sensors for this cost-effective device. A further advantage of this solution is that it requires no physical contact to be made with the liquid medium. Into the bargain we also obtain a tool for measuring distance.

The frequency of 40kHz normally used by ultrasonic transducers corresponds, at an ambient temperature of 20°C, to a wavelength of 8.5 mm.

Features

- Measures the level status of liquids
- Watchdog function monitors fill level, with relay output and alarm LED
- Programming of min./max. alarm levels infinitely variable
- Stores min./max. calibration values for up to ten tanks or containers in memory
- Also measures distance
- Intuitive menu guidance using LCD display

Sound waves in this frequency spectrum do not disperse to any great extent and diverge (in a club shaped form) with an aperture angle of around 15°. Perfect qualifications for our purpose then, since the side walls of the container will have effectively no influence on the reflected signal. Accordingly we will keep the functional description short and to the point. A trigger pulse causes the ultrasonic module to transmit a burst signal. This signal is reflected by the object to be measured, say the water surface in a rainwater barrel, and received back by the ultrasonic module as an echo signal. Since the sound traverses the distance to the reflecting object twice during the echo period, we can state that:

$$\text{distance} = 0.5 \times \text{velocity of sound [m/s]} \times \text{echo period [s]}$$

How temperature influences the velocity of sound

The velocity of sound c_0 in air amounts to 331.5 m/s measured at a temperature of $T = 0^\circ\text{C}$. The velocity of sound is not dependent on atmospheric pressure but purely on the air temperature ϑ in degrees Celsius:

$$c_\vartheta = c_0 \cdot \sqrt{1 + \alpha \cdot \vartheta}$$

where α is the coefficient of expansion of $1/273.15 = 3.661 \times 10^{-3} \text{ } 1/^\circ\text{C}$. From this arises the approximation:

$$c_\vartheta = 20.063 \cdot \sqrt{\vartheta + 273.15} \text{ m/s}$$

We can see that for precision measurement the influence of temperature is by no means insignificant. Here are some examples of sound propagation times at 0°C and 20°C .

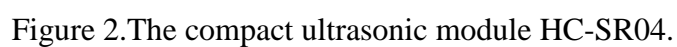
Distance in cm	Time elapsed in ms at 20°C	Time elapsed in ms at 0°C
2	0.117	0.121
10	0.583	0.603
50	2.915	3.017
100	5.831	6.033
200	11.662	12.066
300	17.492	18.100

2. Circuit and Components

The schematic in Figure 1 is based on three components, the well-loved ATmega8 micro-controller, a [4x20 LCD](#) connected to this plus the ultrasonic module. As well as the [HC-SR04](#) module (Figure 2) you can also use the almost identical US-020. Both modules are in the \$5/ £3 price class, making them unbeatable attractive for use in a do-it-yourself measurement device.

Ascertaining the manufacturers of these modules is not an easy task but they are widely available in electronics stores, particularly those that specialize in robotics. When you check out the data sheets, however, you will discover wide variations. One time the detection distance for the two models will be shown as 2 cm–3 m (.8 inch – 10 feet), another time as 3 cm – 7 m (1.2 inch – 23 feet). Further investigation will uncover other divergent details, even for one and the self-same model. Hence the current consumption will vary between 3mA and 15mA depending on which data sheet you read. Reliable data can be determined only by doing your own research. Generally suppliers of these modules agree on the following data:

- Operating voltage: 5VDC.
- Temperature range: $0\text{--}70^\circ\text{C}$.
- Diffusion angle: 15° .
- Operating frequency: 40kHz.
- Trigger input signal: 10µs.



5

highly inflammable fluids, owing to risk of explosion!). If you are taking measurements in containers with particularly small aperture diameters you can resort to the SRF02 ([4] ultrasonic module, which although not pin-compatible, is nevertheless equipped with a serial/I2C interface and uses a single capsule for both transmitting and receiving. As the transmit/receive angle is very acute, with this model you need to bear in mind that the minimum distance from the medium being measured amounts to 16cm. To simplify the menu guidance a four-line, 16-character LCD display is employed. Potentiometer P1 adjusts the contrast. R3 is the series resistor for back-illumination and should amount to around

$$R3 = (5 - Vf) / Inom [\Omega].$$

For this you need to consult the data sheet of the display. With many LCDs the series resistor is already provided internally and in that case you can replace R3 with a wire bridge or strap. To save energy you might also consider a setup involving a switch. The changeover relay RE1, whose contacts are connected to K3, are controlled by port pin PD7 and driver transistor T1, and the alarm LED (low-current) by port pin PD6. A low-dropout voltage regulator LM2940CT-5 ensures a stable 5V supply voltage from four 1.5V batteries. This somewhat exotic voltage regulator delivers up to 1A and can therefore be loaded much more heavily than the usual 78L05 with its 100mA. Naturally you can also use a plug-in ('wall wart') AC power supply (<26VDC) for this project, connected to K1. With the relay energized and the LCD illuminated the circuit draws around 150mA. To achieve maximum accuracy of measurement, the microcontroller needs an external 16MHz crystal. A couple of passive components complete the circuitry.

Components List

Resistor:

R1, R2 = 2.2K Ω R3 = 0 Ω (wire link) R4 = 10k Ω P1 = 10k Ω trimpot

Capacitor:

C1 = 10 μ F 50V C2, C3 = 22pF 50V C4, C7, C8, C9 = 100 μ F 50V, 20% C5, C6 = 47 μ F 50V

Semiconductor:

D1 = 1N4148 LED1 = Red, 3mm T1 = BC547C IC1 = LM2940T-4.0 IC2 = ATmega8-16PU

3. The Control Firmware

The program in the micro-controller must cover the following functions:

- Indication of the level of liquids.
- Watchdog monitoring of this level with relay output and LED alert.
- Infinitely variable programming of the min/max alarm level.
- Storage of min/max calibration values for upto ten containers or tanks.
- Distance measurement.
- Intuitive menu guidance using LCD readout.
- Offset correction for distance measurement.
- Manual divisor correction to accommodate low and high temperature extremes.

The software follows the procedure shown in Figure 3. In view of the limited amount of memory space in the ATmega8, readout messages are given in English only. Operation makes use of the four press buttons Return, Next, Plus and Minus. Because of the complexity of the program coding, which can be downloaded along with the hex file and the PCB layout at [2], it has been subdivided into meaningful procedures and provided with corresponding commentaries, so that even beginners should find their way through it rapidly.

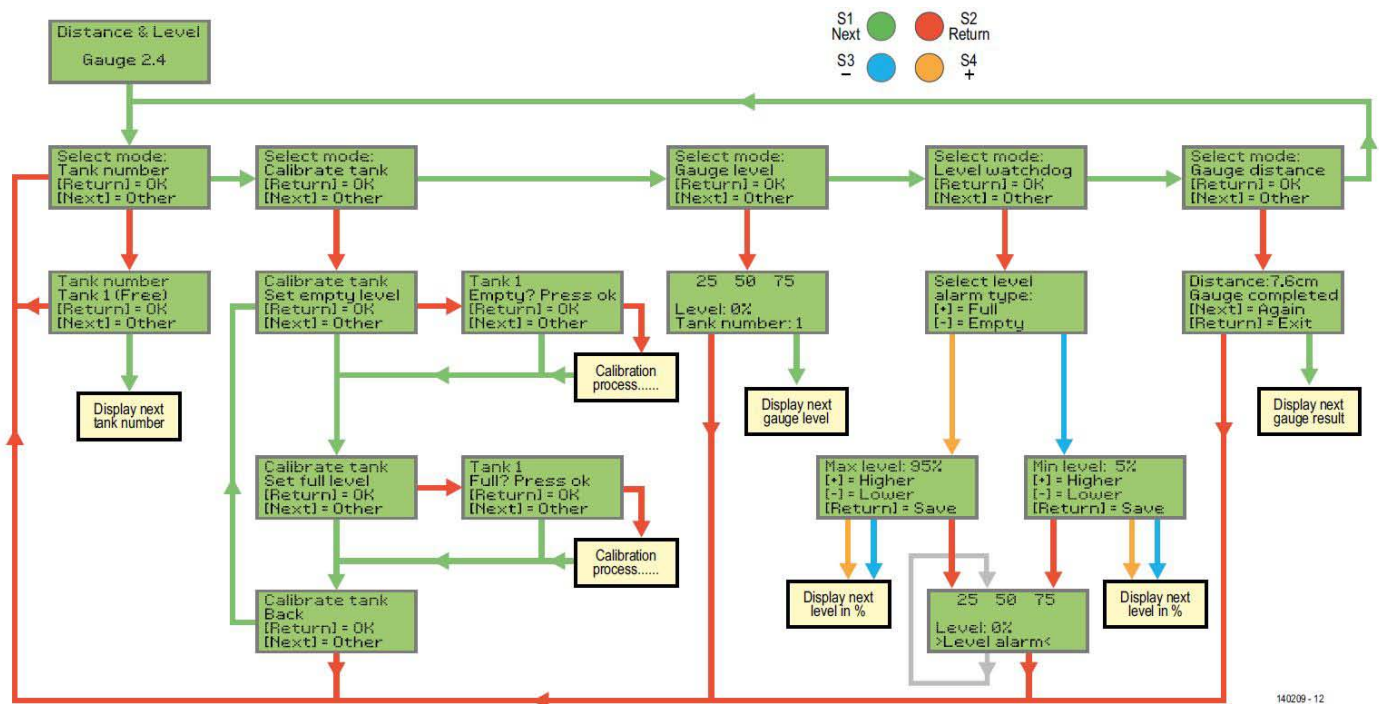


Figure 3. Procedural diagram for the software.

The actual measurement process follows. To bring the ultrasonic module connected to K2 into transmitting a signal, a pulse 10 μ s long with a falling edge is sent via port pin PD2 to the trigger input. As a result, about 250 μ s afterwards, the ultrasonic module transmits a burst signal with a frequency of 40kHz and duration of 200 μ s. The echo output, which is connected to the micro-controller via port pin PD3, switches now to High level and the ultrasonic module stands by to receive the reflected signal. If this occurs, the echo output flips back to Low. Timer1 is ticking away throughout this process, so that after the timer is halted, the distance can be calculated. After 50ms if no echo signal arrives, measurement for the current cycle is cancelled. To achieve the best possible results the mensuration cycle involves 16 separate measurements. The program code for this can be examined in the procedure *Gauge_distance()*.

Now a word of advice for people doing their own 'burning': you can use Arduino Uno board as controller board for this project. Please refer to "Run with Arduino" tab on how to get running with Arduino Board.

4. Construction

Construction is extremely simple thanks to the very manageable number of components. You can buy the double-sided printed circuit board (Figure 4) from Elektor or etch it yourself at home. The relevant files are in the download package [2]. Some handiwork and skill are necessary to create cutouts in the case for the LCD readout, the ultrasonic capsules, the pressbuttons and the switch. How you arrange the ultrasonic capsules is a matter of personal taste. Installing the ultra sonic modules separately on a moderately long screened cable is also conceivable, in case reading needs to take place in another room. Once the

programmed ATmega8 has been inserted and the ultrasonic modules connected, you can switch on the device for the first time. At very least the background lighting for the LCD should illuminate. If this is not the case, disconnect the power supply and double-check the component placing! During the start phase Distance & Level Gauge 2.4 (Figure 5) should be visible on the display for two seconds. Now you can perform the basic settings. To do this you switch off the device, then turn it on again while holding down the Return button. Having released the button, you will be asked the width of the container being measured. At this point you can press the Plus and Minus buttons in order to enter an offset in 5 mm increments for taking into account the distance from the top of the ultrasonic module to the leading edge of the container. This gives you a simple way, for example, of measuring the distance from one wall to the other later on. The best method for this is to take a 2m reference measurement with a two-meter wooden rule (or measure three feet using a carpenter's yardstick). The value is confirmed using Return. Following this you are asked for the divisor value used for correcting the value measured. This is necessary since the velocity of sound is temperature-dependent. In the temperature range from -20°C to $+40^{\circ}\text{C}$ the velocity of sound varies from 312.85 m/s up to 349.32 m/s. The deviation is virtually linear, so that for a temperature variation of 10°C you can expect an inaccuracy of about 2%. Panel 1 contains some formula and a table of values for the elapsed times to be expected. Incidentally, atmospheric pressure has no influence on the velocity of sound.

Anyone prepared to accept up to 5% inaccuracy in this connection need do nothing at all. On the other hand, if you wish to take measurements in temperatures below 15°C or above 25°C and with a higher level of accuracy, a division factor must be introduced. The higher the temperature, the greater the divisor must be. The values for the width of the container and the divisor must also be retained after a battery change and can be entered afresh at any time. The *Select* mode: menu should now be visible. Pressing the Return button will take you to the tank selection menu. The opportunity is provided to carry out calibration functions for up to ten different tanks or other containers. The only condition made is that the container shape must be as close as possible to a cuboid or cylindrical form, otherwise linear measurement is impossible. Having selected your tank number you land back at the main menu. After pressing the Next button the menu option Calibrate tank appears. Hitting Return now goes to the calibration menu. Having selected empty-level calibration, you are asked if you are ready to perform the calibration. You don't necessarily need to empty the vessel in question for this purpose; instead you can simply measure outside the tank the height from the filler neck down to the ground.

Once the empty level has been calibrated, you reach the menu for full-level calibration (Figure 6). The default value is set as a minimum of 3cm, which you can accept by pressing Next. If, however, you wish to perform a calibration for the full-level, put the device in position above the filled container and press Return. After returning to the main menu you can perform the first level measurement. To do this, select menu option Gauge level and confirm with Return. Measurement begins now and the result is displayed. Next carries out a fresh measurement, Return leads back to the main menu. To test out level monitoring you should select the Level Watchdog menu. In the menu that follows (Figure 7) the alarm type can be selected: Full will activate the alarm when the permitted fill quantity is exceeded and Empty when it falls short. Having chosen your alarm type, you can alter the threshold value in the Level Alarm menu (Figure 8). For the default values we have defined the maximum level as 95% and the minimum level as 5%, but these can be varied using the Plus and Minus buttons. All the calibration values made are stored in the EEPROM and are retained there even after a change of battery. The first measurement follows next and the current level is displayed (Figure 9). Measurements are now taken every five seconds. If these fall below the minimum value or exceed the maximum value set, the display shows > Level alarm <, the red LED illuminates and the relay operates, for example to switch on a pump. If the level status returns into the permitted range after 5s, the alarm function is cancelled. The 5s measurement cycle avoids the need for a hysteresis function to eliminate relay contact bounce. To exit from this mode back to the main menu, press the Return button (for up to 5s). You can now select the Gauge distance menu. Each press on the Next button initiates a distance measurement (Figure 10). As before, you can exit from this mode and go back to the main menu by pressing Return.



Figure-5

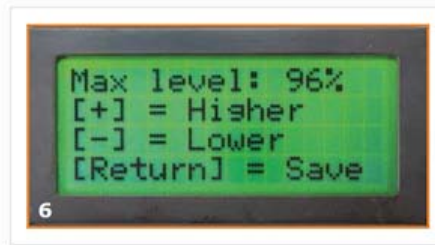


Figure-6



Figure-7

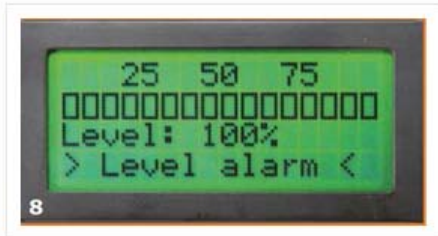


Figure-8

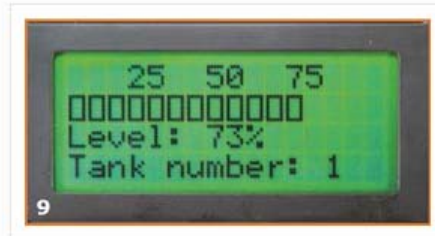


Figure-9

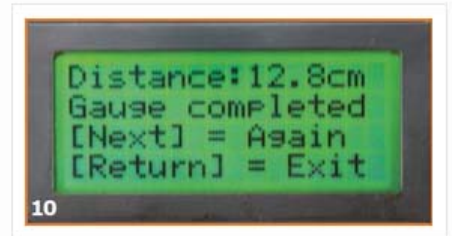


Figure-10

5. Using Bascom-AVR with Arduino

The ARDUINO is a hardware platform based on AVR processors. ARDUINO boards/chips are programmed with a bootloader. This bootloader is the old STK500 protocol, not longer supported by Atmel in Studio. There are various programmers for ARDUINO, AVRDUDE is probably the most versatile. BASCOM also supports the ARDUINO/STK500 v1 protocol. The DTR/RTS lines are used to reset the board. You can program/read flash/EEPROM but you cannot read/write fuse/lock bytes. The STK500 bootloader for ARDUINO does not support this.

Under options you only need to select the programmer, and the COM port. Since an FTDI chip is used on most ARDUINO boards, this is a virtual COM port. Only present when the USB cable is connected to your PC.

Select 57600baud for the baud rate. Older ARDUINO boards work with 19200baud. Optiboot Bootloader (under Windows 7)

ARDUINO V2:

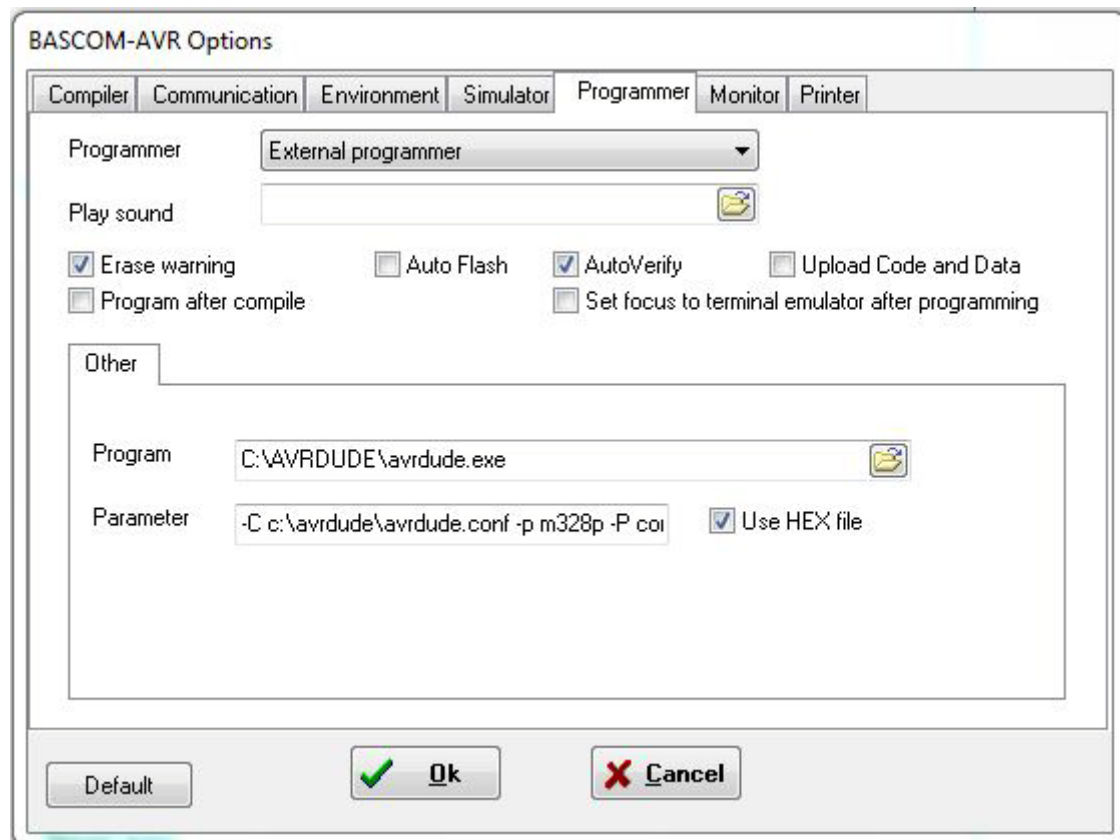
The developers of the ARDUINO finally implemented the STK500V2 protocol. This protocol is supported by Atmel and of course by BASCOM. Select the ARDUINO STK500V2 programmer in BASCOM programmer options to use this protocol. A board like the MEGA2560 R3 uses this protocol and probably all newer AVR based ARDUINO boards will support this protocol. The baud rate should be 115200 but could be different for your board.

Using Bascom-AVR with Arduino Optiboot Bootloader (under Windows 7):

For more information on Optiboot visit following website: <http://code.google.com/p/optiboot/>

1. Download AVRDUDE from <http://www.nongnu.org/avrdude/>
2. Latest Windows Version (April 2012): avrdude-5.11-Patch7610-win32.zip. Complete link:
 1. <http://download.savannah.gnu.org/releases/avrdude/avrdude-5.11-Patch7610-win32.zip>
3. Create a folder like c:AVRDUDE

4. Copy the content of avrdude-5.11-Patch7610-win32.zip in this new folder
5. Open Bascom-AVR
6. Click on Options >>> Programmer
7. Choose External programmer
8. Checkmark Use HEX file
9. Include the path to avrdude.exe
10. User Parameter: **-C c:avrdudeavrdude.conf -p m328p -P com19 -c arduino -b 115200 -U flash:w:{FILE}:i**



Explanation of Parameter:

-C

c:avrdudeavrdude.conf The config file tells avrdude about all the different ways it can talk to the programmer.

-p

m328p This is just to tell it what microcontroller its programming. For example, if you are programming an Atmega328p, use m328p as the partnumber

-P

com19 This is the communication port to use to talk to the programmer (COM19) in this case. Change it to your COM port.

-c

arduino

Here is where we specify the programmer type, if you're using an STK500 use stk500, use arduino for Optiboot

-b

115200

Set serial baudrate for programmer. Use 115200 baud for Optiboot.

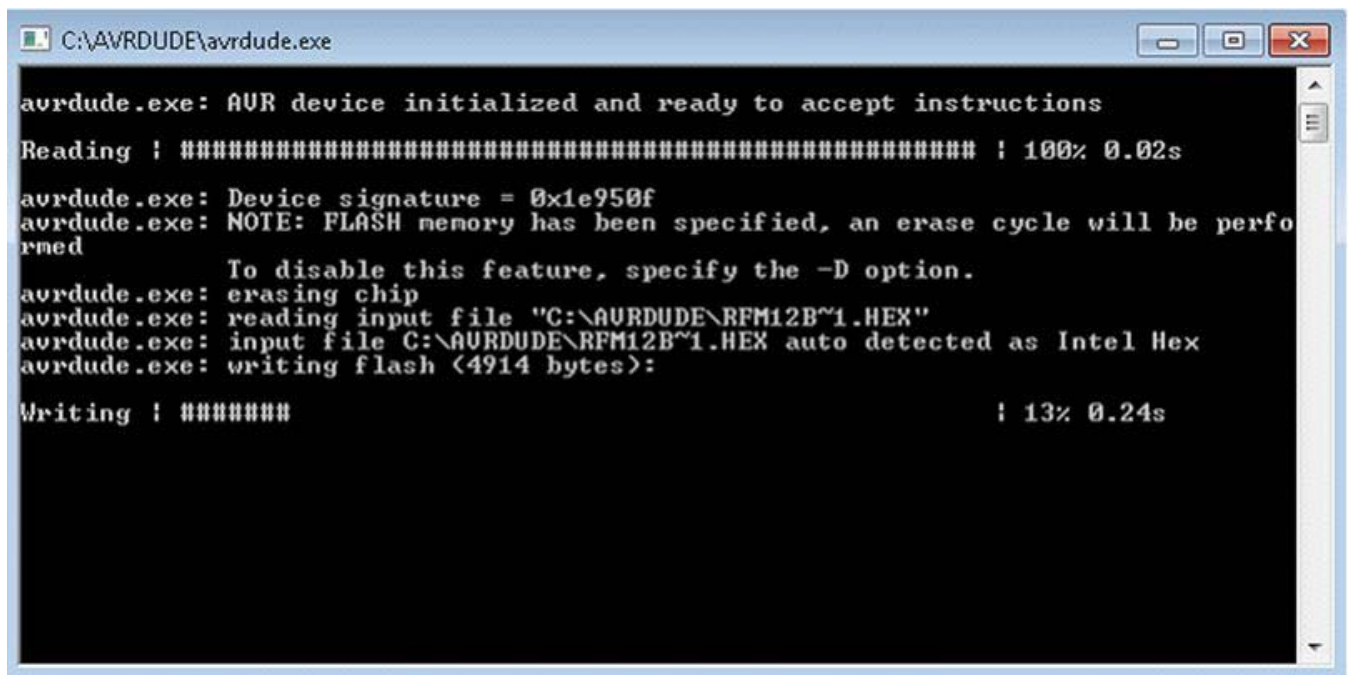
-U

flash:w:{FILE}:i

You define here:

- the memory type: flash or eeprom (this could be also hfuse, lfuse or effuse if you want to verify this)
- r (read), w (write) or v (verify)
- Use {FILE} to insert the filename {EEPROM} to insert the filename of the generated EEP file.
- i = Intel Hex File

After clicking on the F4 (Program Chip) Button in Bascom-AVR you see the CMD window of Windows 7 until AVRDUDE is ready flashing the Arduino.



```
C:\AVRDUDE\avrdude.exe

avrdude.exe: AVR device initialized and ready to accept instructions
Reading : ##### : 100% 0.02s
avrdude.exe: Device signature = 0x1e950f
avrdude.exe: NOTE: FLASH memory has been specified, an erase cycle will be performed
To disable this feature, specify the -D option.
avrdude.exe: erasing chip
avrdude.exe: reading input file "C:\AVRDUDE\RFM12B~1.HEX"
avrdude.exe: input file C:\AVRDUDE\RFM12B~1.HEX auto detected as Intel Hex
avrdude.exe: writing flash (4914 bytes):
Writing : ##### : 13% 0.24s
```

[Complete documentation of AVRDUDE parameters](#)

<http://avrhelp.mcselec.com/index.html?arduino.htm>

-
- [BASCOM-AVR2.0.7.5](#)
 - [Ultrasonic Level Meter Source Code](#)

6. Key Components

1. [Arduino Uno](#)
2. [HC-SR04 Ultrasonic Module](#)
3. [GDM2004D-FL-YBS : 20×4 LCD Panel](#)
4. [Mechanical Relay](#)