

ATmega 2560

In the d-diot board the function of the ATmega2560 microcontroller is to process the signals of the commercial devices that communicate at 433 Mhz.

This is done running the [RFLink](#) firmware on the microcontroller in association with the RXB6 and FS1000A receiver and transmitter modules. Of course you can run any sketch you want on the ATmega2560; the instruction to upload the RFLink firmware are provided [here](#), while if you want to upload a custom sketch (ino file), go [here](#).

The serial connection between the ATmega2560 chip and the Raspberry Pi is through the pins 2 and 3 of the microcontrollers and the GPIO pins 14 and 15 of the Raspberry Pi.

The autoreset function when uploading a sketch is provided by the GPIO pin 13 of the Raspberry Pi, bridging the P33 and RST pin of the JP2 jumper. The manual change of the JP2 jumper is required because the GPIO pin 13 (pin number 33) is shared and normally it provides the IRQ functionality of the MySensors nrf24 gateway.

Autoreset hack

The autoreset function in the d-diot board is possible thanks to the great hack of deanmao to the avrdude program (see [here](#) and [here](#)) and to the optimized code provided by spellfoundry (see [here](#) and [here](#))

To enable the autoreset follow the step below, starting with the creation of the directory tree.

```
pi@d-diot:~ $ mkdir /home/pi/atmega2560
pi@d-diot:~ $ mkdir /home/pi/atmega2560/arduino-mk
pi@d-diot:~ $ mkdir /home/pi/atmega2560/rflink
pi@d-diot:~ $ mkdir /home/pi/atmega2560/autoreset
```

Download the file from spellfoundry repo:

```
pi@d-diot:~ $ cd /home/pi/atmega2560/autoreset
pi@d-diot:/home/pi/atmega2560/autoreset $ git clone
https://github.com/SpellFoundry/avrdude-rpi.git
```

Change the raspberry GPIO PIN in the autoreset file. Set it to GPIO pin 13 (line 16); the default value is 22.

Change also the time sleep from 0.32 to 0.16 (line 19)

```
pi@d-diot:/home/pi/atmega2560/autoreset $ nano
/home/pi/atmega2560/autoreset/avrdude-rpi/autoreset
```

[/home/pi/atmega2560/autoreset/avrdude-rpi/autoreset](#)

```
#!/usr/bin/python

import RPi.GPIO as GPIO
import sys, os, re, time, fcntl

fd = sys.stdin.fileno()
fl = fcntl.fcntl(fd, fcntl.F_GETFL)
fcntl.fcntl(fd, fcntl.F_SETFL, fl | os.O_NONBLOCK)

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
dtr = re.compile('.+TIOCM_DTR.+')
start = time.time()

def reset():
    pin = 13
    GPIO.setup(pin, GPIO.OUT)
    GPIO.output(pin, GPIO.HIGH)
    time.sleep(0.16)
    GPIO.output(pin, GPIO.LOW)

def process():
    while True:
        try:
            duration = time.time() - start
            input = sys.stdin.readline().strip()
            if dtr.match(input):
                reset()
                return
            elif duration > 5000:
                return
        except:
            continue

process()
GPIO.cleanup()
print "done with autoreset - ignore Broken pipe errors if Done
uploading"
exit
```

Copy the autoreset and avrdude-autoreset file in the /usr/bin directory:

```
pi@d-diot:/home/pi/atmega2560/autoreset $ sudo cp
/home/pi/atmega2560/autoreset/avrdude-rpi/autoreset /usr/bin
pi@d-diot:/home/pi/atmega2560/autoreset $ sudo cp
/home/pi/atmega2560/autoreset/avrdude-rpi/avrdude-autoreset /usr/bin
```

Backup the original avrdude file:

```
pi@d-diot:/home/pi/atmega2560/autoreset $ sudo mv /usr/bin/avrdude
/usr/bin/avrdude-original
```

Symlink avrdude-autoreset to avrdude

```
pi@d-diot:/home/pi/atmega2560/autoreset $ sudo ln -s /usr/bin/avrdude-
autoreset /usr/bin/avrdude
```

Arduino-mk

The arduino-mk is the program that allows you to compile and upload any sketch (.ino) from your Raspberry Pi to the ATmega2560 microcontroller, without the requirements of an Arduino IDE. The upload is done through the avrdude program, so the autoreset functionality is still present.

The arduino-mk has more capabilities respect to those discussed in this wiki: see [here](#) and [here](#) if you want to learn more.

To use the arduino-mk you need a Makefile:

```
pi@d-diot:~ $ nano /home/pi/atmega2560/arduino-mk/Makefile
```

Add the following lines:

[/home/pi/atmega2560/arduino-mk/Makefile](#)

```
ARDUINO_DIR = /usr/share/arduino
BOARD_TAG   = mega2560
ARDUINO_PORT = /dev/serial0
ARDUINO_LIBS =
include /usr/share/arduino/Arduino.mk
```

Exit (CTRL+x) and save the changes (y).

Create an empty arduino sketch to test arduino-mk:

```
pi@d-diot:~ $ nano /home/pi/atmega2560/arduino-mk/test.ino
```

Add the following lines:

[/home/pi/atmega2560/arduino-mk/test.ino](#)

```
//Test
void setup()
{
}
void loop()
```

```
{  
}
```

Exit (CTRL+x) and save the changes (y).

Usage

Follow this guides to use your ATmega2560:

- [Upload RFLink firmware](#)
- [Upload a custom sketch](#)
- [Serial communication using minicom](#)

From:
<https://wiki.d-diot.com/> - **d-diot wiki**

Permanent link:
https://wiki.d-diot.com/system_administration/manual_installation/4_atmega2560

Last update: **2019/04/14 09:54**

