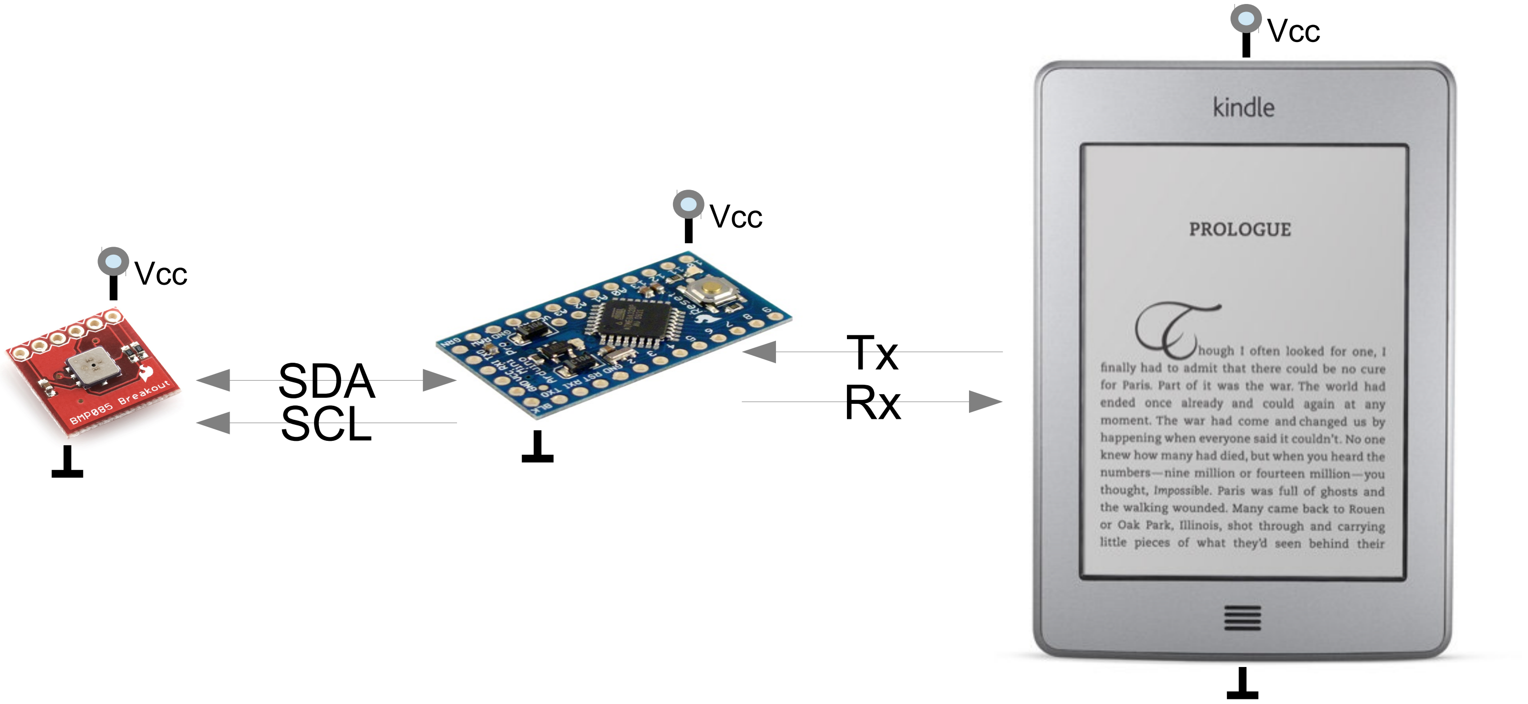
Kindle Touch® Personal Weather Station

by Leonard Pitzu
mail to: leonard.pitzu@gmail.com

This modd will extend the Kindle Touch by adding ***some hardware*** which will provide a way of interfacing the Kindle with some sensors.


Vcc
Vcc
Vcc
SDA
SCL
Tx
Rx
BMP085 Breakout
kindle
PROLOGUE
Though I often looked for one, I finally had to admit that there could be no cure for Paris. Part of it was the war. The world had ended once already and could again at any moment. The war had come and changed us by happening when everyone said it couldn't. No one knew how many had died, but when you heard the numbers—nine million or fourteen million—you thought, Impossible. Paris was full of ghosts and the walking wounded. Many came back to Rouen or Oak Park, Illinois, shot through and carrying little pieces of what they'd seen behind their

DISCLAIMER: *IF YOU WILL NOT BE CAREFULL THIS CAN BREAK YOUR KINDLE TOUCH AND WILL VOID THE WARRANTY. I AM NOT RESPONSIBLE FOR ANY BRICKED DEVICES OR ISSUES YOU ENCOUNTER. I WILL PROVIDE EVERYTHING AS IT IS AND GUIDE EVERYBODY THOURGH DOING THIS MODD BUT THIS IS WHERE MY HELP ENDS. YOU NEED BASIC HARDWARE AND SOFTWARE KNOW HOW AS WELL AS WILLINGNESS TO SPEND A FEW HOURS DOING THIS MODD.*

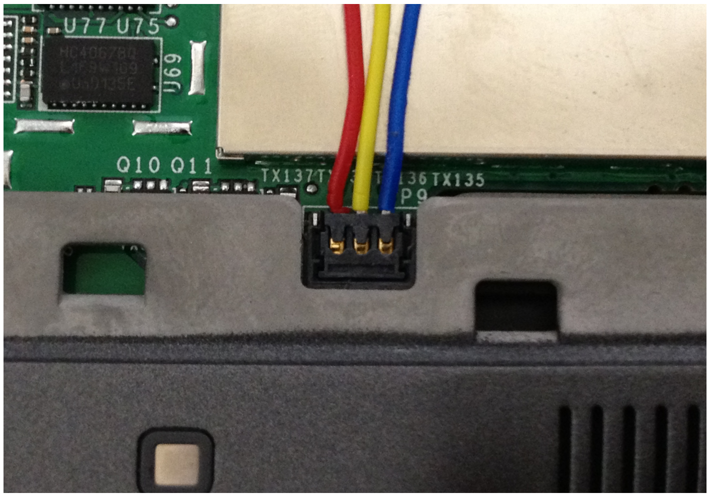
Needed tools:

- jailbroaken Amazon Kindle Touch WiFi only version (this is the one I have and the one that has enough room inside the case to fit all the extra electronics)
- GUI launcher installed
- a 3.3V usb-to-serial converter (i use this one with the Rx and Tx Led’s removed)
- an *Arduino Pro Mini* board (modified as it will be detailed below), 3.3V version
- a sensor (in this case I use the I^{2}C BMP085 pressure/temperature sensor – this one is quite reliable and low power!!!)
- lots of electronics stuff (1 transistor, some resistors, a capacitor, wires, etc.)
- patience :-)

In order to do this following steps will be needed:

1. ***Disable the serial console.***

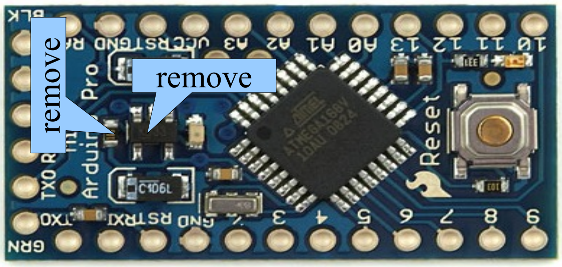
   By default the console is routed to the serial port (ttymxc0). In order to use the serial port for plain text data communication the console needs to be disabled. To do this connect take the usb-to-serial converter and remove the Led's from the Rx and Tx line. Next solder 3 wires to the adapter: Rx, Tx and GND.
   In the picture below (taken from http://www.flickr.com/photos/hondamarlboro/6466353083/in/set-72157628233938657/) the serial pins on the Kindle Touch are located, from left to right: GND, Rx and Tx.



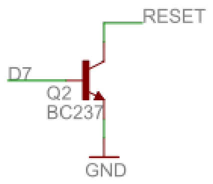
   On the PC start Putty or any other terminal software you like. Connect to the Kindle using 115200, 8, n, 1. Now log into root (username: *root* password: *fiona023*). Now just move /etc/init/console.conf to /mnt/us. After that, if You reboot Your Kindle YOU WILL LOOSE YOUR CONSOLE. There is no way of getting it back except for running a WAF application which restarts the console so that You can copy console.conf back to /init. Be carefull – removing the console is dangerous; this is the main back door to the Kindle!!!

2. ***Modify the Arduino Pro Mini board.***

   The board comes with a power supply, some Led's and other "useless" parts. All this need to be removed in order to reduce the power consumption of the board and make it "slim and fit".

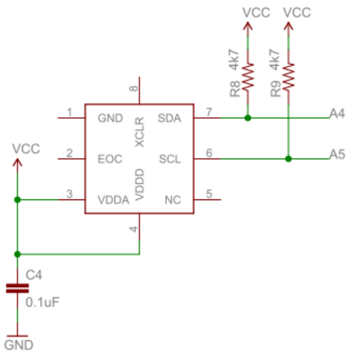


   The board is now slim but it's not complete. Arduino's come with a bootloader which prevents, by default, the use of the watchdog. To overcome this, without modifying the bootloader (I have no programmer at hand this is why I am doing this modification) i've added a transistor connected to the reset pin, as follows:



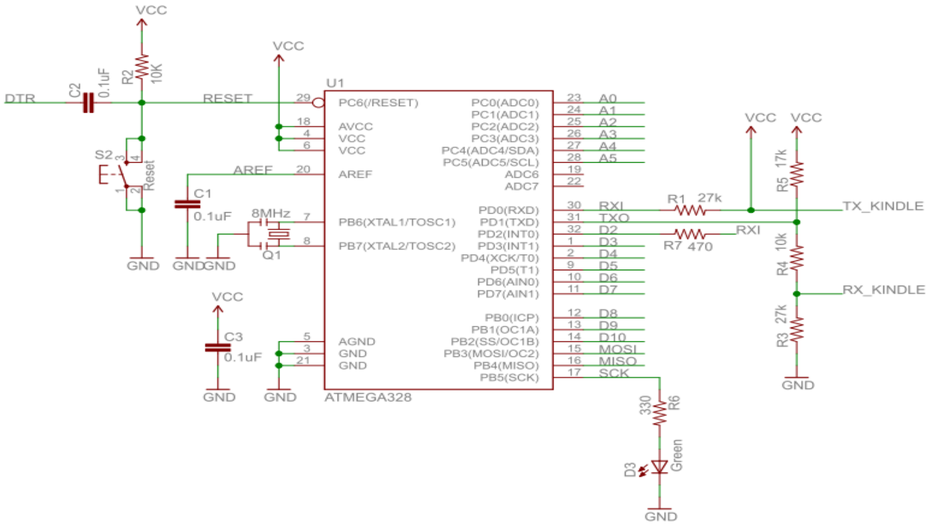
When pin D7 goes high the board is reset. It’s ugly but it does the trick.

3. ***Hook up the sensor to the Arduino board (and check that it works)***.



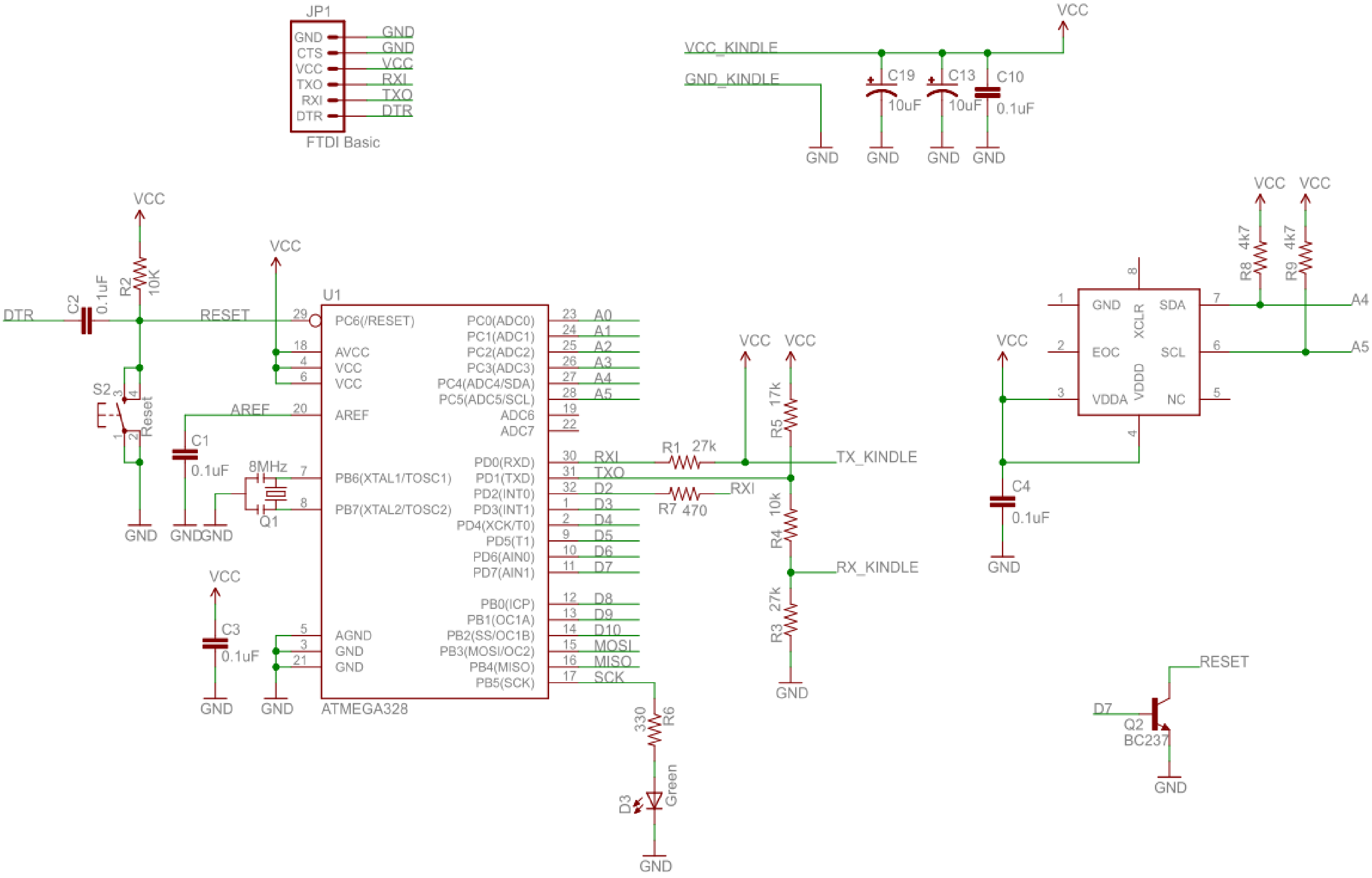
The sensor is connected to the Vcc (3.3V), GND, SDA (pin A4) and SCL (pin A5) of the Arduino board. It’s a low power I^{2}C chip which measures the pressure and temperature. You can write Your own software or use one of those libraries around and test the sensor by running the demo sketch that comes with the library. If the output is successful, like in the linked tutorial then You can move to the next step.

4. ***Hook up the board + sensor to the Kindle.***



The Kindle’s serial interface is 1.8V and the Arduino is 3.3V. Hooking them together without some means of adapting the logic levels would fry the Kindle. R1, R3, R4 and R5 are used to adapt the logic levels so that the communication can take place. In operation mode the Arduino board is in power down. This provides a huge power saving but there is no other way to wake up the chip as the main clock source is turned off. Using R7 the Rx pin is connected to an external interrupt. During power down only asynchronous interrupt sources are available. Thus, when the first bit arrives from the Kindle, on the Rx line, INT0 (pin D2) is triggered and the chip wakes up. D3 lights up as the chip boots and indicates that “something is happening”. The Led light only for 50ms (Led are very power hungry – maybe this should be removed completely).

The complete schematic looks like this:



5. ***Program the board.***

Using the same usb-to-serial converter as before, load the sketch from the attached archive, using the Arduino IDE.

6. ***Install the WAF.***

From the archive copy the folder named *weatherconsole* to the extension folder on Your Kindle. After You disconnect the Kindle make sure You reboot it so You can see the app in the menu.

7. ***Test the system.***

Start the app and see if anything is displayed. Maybe it will not work from the first try so You could run it a few times. If it does not work... use the mail above to ask & blame me :-D

Measured power consumption on the Arduino Pro Mini board fitted inside the Kindle Touch:

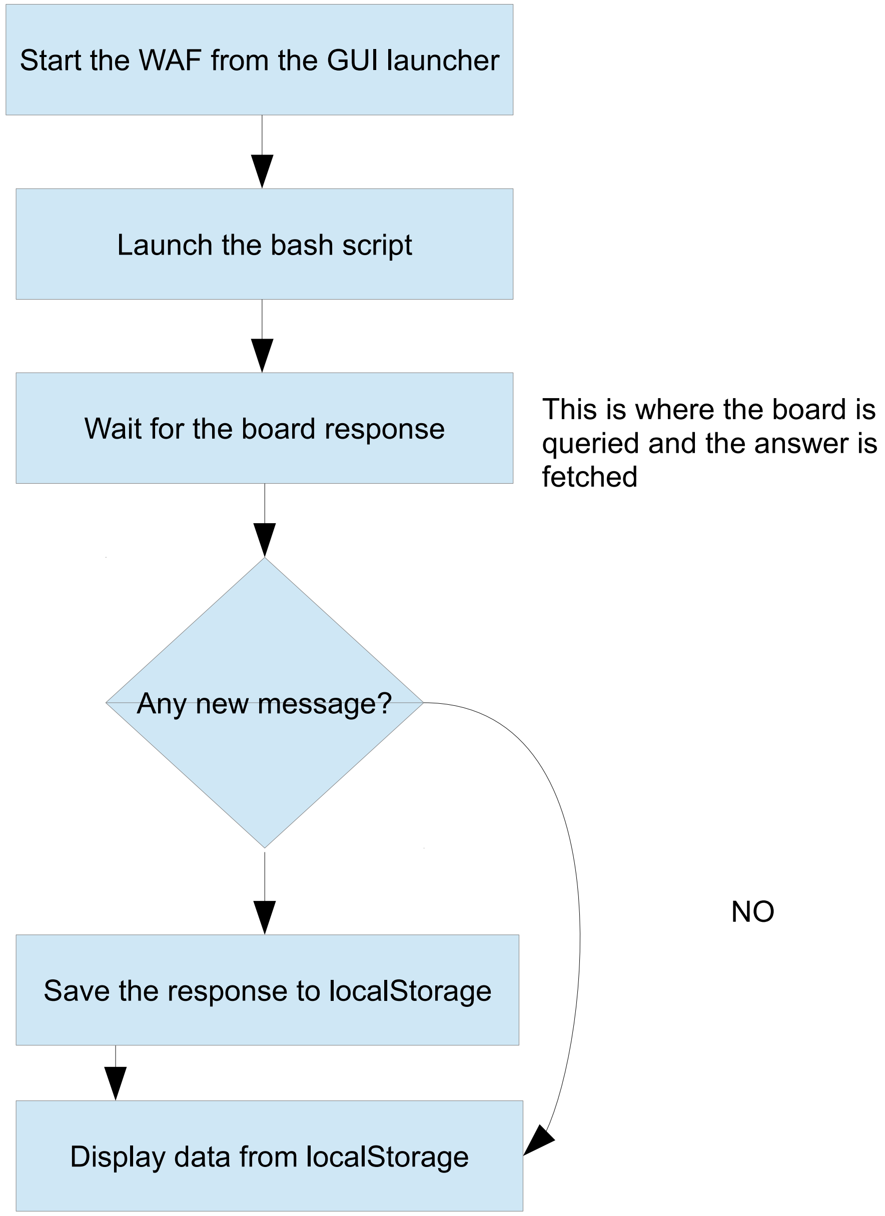
RUNNING (pwr supply)	: 19 mA
RUNNING (3.3V direct)	: 17 mA
SLEEP_MODE_PWR_DOWN	: 15.0 mA

atmega328 (arduino pro mini with pwr. supply & pwr. led removed)

RUNNING	: 15.5 mA
SLEEP_MODE_IDLE	: 15 mA
SLEEP_MODE_ADC	: 6.5 mA
SLEEP_MODE_PWR_SAVE	: 1.62 mA
SLEEP_MODE_EXT_STANDBY	: 1.62 mA
SLEEP_MODE_STANDBY	: 0.84 mA
SLEEP_MODE_PWR_DOWN	: 360 uA
SLEEP_MODE_PWR_DOWN (BROWN OUT disabled)	: 335 uA
SLEEP_MODE_PWR_DOWN (ADC disabled)	: 0.335 uA
real board consumption (external int set up - pwr. Down)	: 0.3 mA
real board consumption (active incl. 1 led for 50ms)	: 3.7 mA

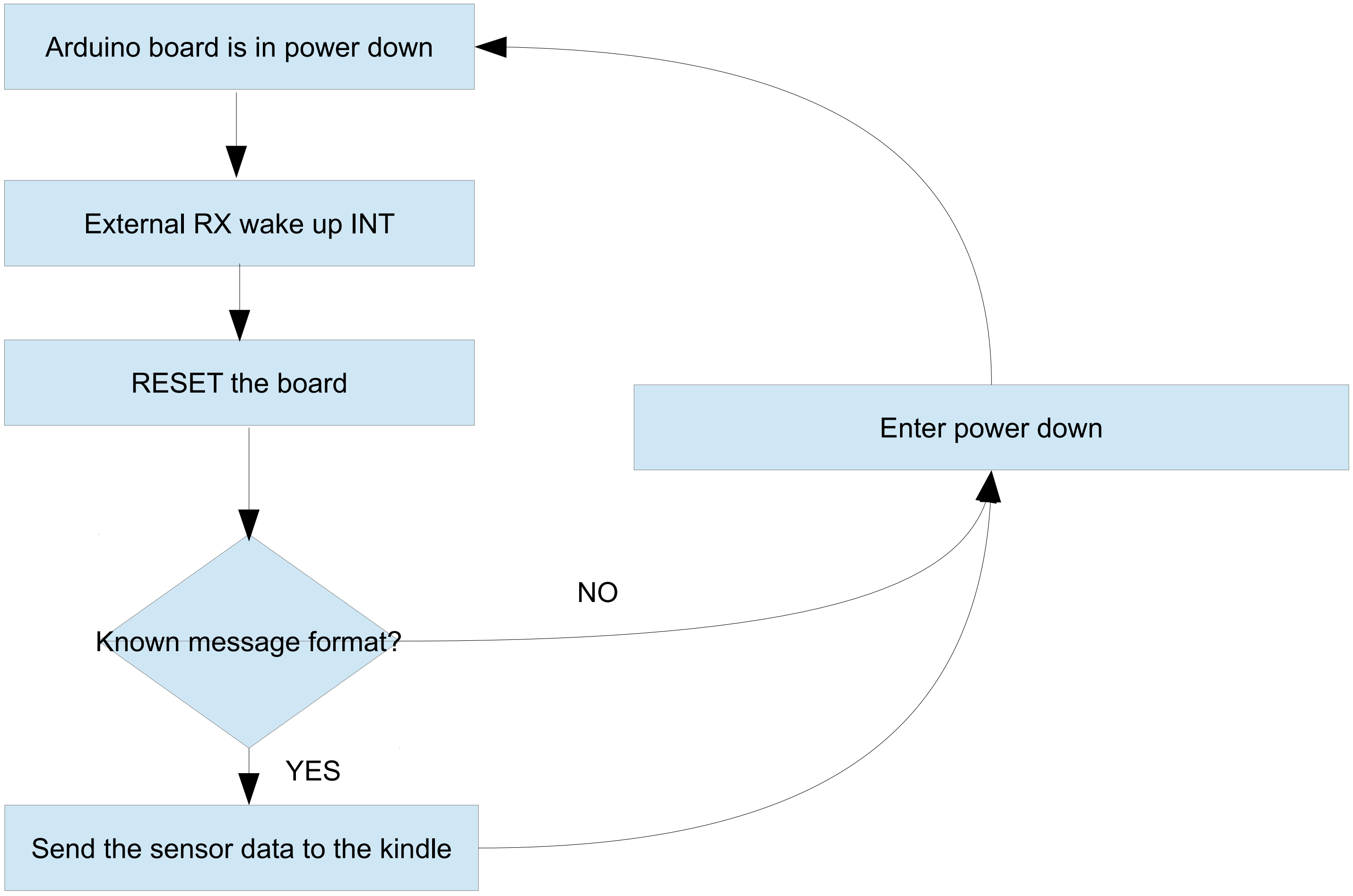
This is the final result. This is what the board draws from the Kindle.

The basic flow is:


Start the WAF from the GUI launcher
Launch the bash script
Wait for the board response
Any new message?
NO
Save the response to localStorage
Display data from localStorage

This is where the board is queried and the answer is fetched

application flow



Arduino board flow