Instructions for Reproducing the Flowboard

These instructions are meant to allow those interested in building the Flowboard themselves to do so. As the Flowboard is only a first prototype, the design is by no means perfect with many hacky, imperfect solutions. The user of these instructions is invited to optimize the design where he/she wishes.

Components

The following hardware components are necessary to build the Flowboard:

- 1x iPad (iPad Pro 12.9")
- 1x Arduino Uno Rev3
- 1x Atmega328P
- 18x ADG719BRTZ-REEL7
- 2x HM-10 Bluetooth Low Energy module
- 2x breadboard (with 63 rows)
- 36x green LED (Kingbright L-2060GD)
- 36x resistor (150 Ohm, SMD 0603 package)
- 1x single pole switch (snap-in, rocker switch)
- 1x two pole switch (snap-in, rocker switch)
- 1x 6 AA battery holder (SBH 361A)
- 6x 1.5 V AA battery
- 1x open ended cable with male barrel connector, 5.5 mm OD, 2.1 mm ID
- 1x 18x2 male pin header
- 2x 18x1 male pin header
- 1x 10x1 male pin header
- 2x 8x1 male pin header
- 1x 6x1 male pin header
- 1x 4x2 male pin header
- 2x 4x1 male pin header
- 1x 2x2 male pin header
- 37x 2x1 male pin header
- 2x 1x1 male pin header
- 4x 18x1 female pin header
- 2x 4x1 female pin header
- 43x 2x1 female pin header
- lots of wires

There is also a bill of materials provided with these instructions with links for buying the components and prices included.
Hardware Configuration

Before starting with soldering and putting everything together, the HM-10 BLE modules and the Atmega328P microcontroller need to be flashed with the correct software and configured appropriately.

HM-10

The HM-10 modules are used for the communication between the iPad and the microcontrollers using Firmata. One is used for the communication with the Arduino Uno, the other for communicating with the additional Atmega328P. To allow a fast serial communication over bluetooth between the iPad and the microcontrollers, a few changes need to be made to the standard configuration of the HM-10 modules. Per default, the HM-10 has only one characteristic which can be used for both transmitting and receiving data. We want to enable a second characteristic to have separate characteristics for TX and RX. This also allows us to set a much higher baud rate than the default 9600 Bd without dropping data. Lastly, at least one of the HM-10 needs to be configured with UUIDs for service and characteristic different than the default ones so that both modules have unique UUIDs.

First, check the firmware installed on the HM-10 using the **AT+VERR?** command. Martyn Currey provides nice instructions on how to talk to the HM-10 using either an **Arduino** or a **USB to serial UART adapter**. You need **at least firmware version 5.45** to enable the second characteristic. If you have an older firmware installed, update to the latest firmware using **Martyn Currey’s instructions** (you will need a Windows PC and a USB to serial UART adapter to update the firmware).

If the firmware is new enough, enable the second characteristic by issuing the **AT+FFE21** command. Next, set the baud rate to the maximum of **115200 Bd** using the command **AT+BAUD4**. The steps so far need to be performed for both HM-10 modules. What's left is to do is to assign unique UUIDs for the two modules. For the sake of simplicity, we use the default UUIDs for one of the modules. This will be the one which communicates with the Arduino Uno. Rename the UUIDs of the other module with the commands **AT+UUID0xAAE0** (set service UUID to 0xAAE0) and **AT+CHAR0xAAE1** (set UUID of first characteristic to 0xAAE1, the second characteristic automatically assumes the UUID 0xAAE2).

Atmega328P

The Flowboard contains an Atmega328P microcontroller additional to the one of the Arduino Uno. This secondary microcontroller is used to toggle the ADG719 analogue switches that connect the Arduino pins to either the input or the output side of the Flowboard. In the code of the Flowboard iPad app, the switches are set appropriately every time the mode of an Arduino pin is adjusted by setting the corresponding pin on the Atmega328P to high or low. To allow this, the Atmega328P also runs Firmata and therefore can be handled the same
way as the Arduino Uno. To get Firmata onto the Atmega328P, we burn the Arduino bootloader onto the microcontroller. To do so, follow the instructions (for the minimal circuit setup) from the Arduino website. The instructions also show how to wire up the Atmega328P with an Arduino to upload sketches to the Atmega328P. Load the StandardFirmata.ino sketch onto the Atmega328P. You can find it in the Arduino IDE under File -> Examples -> Firmata -> StandardFirmata. In the setup function of the sketch, you can set the baud rate used by firmata by setting the parameter of the Firmata.begin() line. Don’t forget to set it to 115200 to match the HM-10 module. There is a simple test application if you want to test out whether Firmata was installed correctly. Be aware that the test application only works for a baud rate of 57600 Bd, which is the default value set in the StandardFirmata sketch.

Hardware Assembly

The Flowboard circuitry consists of three boards, for all of which the Eagle files are provided with these instructions: The main board (IO-switching) and the boards which contain the input (breadboard-adaptor-in) and output pins (breadboard-adaptor-out). The two latter ones are visible to the user, while the main board is hidden away in the casing.

The input and output boards hold the pinouts which the user connects his/her hardware components to. These pinouts are connected through wires and switches with the pins of the Arduino on the main board. Additionally, the boards have one LED per pinout as a visual feedback of the pin status for the user. Each available pin of the Arduino is doubled on the Flowboard - there is one pinout for each pin on both the input and the output side - but a pin cannot be connected with both the input and the output simultaneously. If an LED is on, it signals that the corresponding pinout is currently connected with the Arduino pin. The corresponding LED on the other side should be off. That pinout is floating and attaching something to it has no effect.

The main board is responsible for toggling the switches so that the Arduino pins are connected to the pinouts on the correct side of the Flowboard. The switches are controlled by the secondary MSP328P. Each switch is controlled by the pin on the helper microcontroller which has the same Arduino number as the Arduino pin that the switch is connected with. The second task of the main board is to provide the supply voltage for all the parts of the Flowboard (HM-10 modules, MSP328P helper microcontroller, LEDs, breadboards). The supply voltage comes from the 5 V output of the Arduino Uno, which itself is powered by either 6 AA batteries in series or via USB.

The separate boards of the Flowboard are connected with pin headers and wires soldered to them. This approach leads to a ton of soldering and is susceptible to broken connections, but it allows the parts to be plugged in and out easily and made it easier to create a first version of the casing since the lack of rigid connections lowered precision requirements. The placement of the parts on the boards should be understandable with the Eagle files alone. The connections between the boards are probably less self-explanatory and will be explained below.
Main Board

This is the board view of the main board. The numbered red boxes indicate patches which are explained one by one below. All the pin headers detailed below are to be inserted on the top side of the PCB.
Male pin header

This will be connected to the input pinouts through wires soldered to an 18x1 female pin header

Male pin header

This will be connected to the output pinouts through wires soldered to an 18x1 female pin header

Male pin header

This will be connected to the LEDs on the input and output boards with two 18x1 female pin headers with wires soldered to them, one each for input and output

Male pin header

This is connected to the HM-10 module that communicates with the Uno with a 4x1 female pin header with wires soldered to it. The other endings of the wires are soldered directly onto the HM-10. The same applies for the other HM-10 in the lower right corner of the main board. Zoom in on the board for the pin assignments.

Male pin header

Again, a 2x1 female pin header with wires soldered to them will be plugged into this. More details on the connection are in the section about power supply
Input and Output Boards

Again, first an overview of the two boards. The numbered patches are detailed below.
A 1x1 male pin header, inserted on the bottom side of the board. The header is soldered directly to a wire, the other end of which is soldered to a 2x1 female pin header. That female pin header is connected to VCC, which is explained in more detail in the section about the power supply.

(a) Since there doesn’t seem to be 1x1 female pin headers, take a 2x1 female pin header and snap it into ideally two 1x1 headers, but usually one of the two is damaged in a way that it’s not usable. Insert a 1x1 female pin header on the top side of the board. This is the input pinout that the user will use.

(b) The LED for the visual feedback, inserted on the top side of the board. Since the LED is placed between VCC and the Atmega328P (via (c)), it will light up whenever the Atmega outputs low on that pin, which is what happens when the corresponding analogue switch is toggled to the input pinout.

(c) A 2x1 male pin header is inserted on the bottom side of the board for (c) and (d) together. (c) is directly soldered to a wire, of which the other end is soldered to the 18x1 female pin header belonging to the input strip in patch 4 of the main board.

(d) Directly soldered to a wire, of which the other end is soldered to the 18x1 female pin header belonging to patch 2 of the main board.
A 1x1 male pin header, inserted on the bottom side of the board. The header is soldered directly to a wire, the other end of which is soldered to a 2x1 female pin header. That female pin header is connected to GND, which is explained in more detail in the section about the power supply.

(a) A 1x1 female pin header on the top side of the board. This is the output pinout that the user will use.
(b) The LED for the visual feedback, inserted on the top side of the board. Since the LED is placed between GND and the Atmega328P (via (c)), it will light up whenever the Atmega outputs high on that pin, which is what happens when the corresponding analogue switch is toggled to the output pinout.
(c) A 2x1 male pin header is inserted on the bottom side of the board for (c) and (d) together. (c) is directly soldered to a wire, of which the other end is soldered to the 18x1 female pin header belonging to the output strip in patch 4 of the main board.
(d) Directly soldered to a wire, of which the other end is soldered to the 18x1 female pin header belonging to patch 3 of the main board.
Power Supply

This is the section where all the parts of the Flowboard are connected to the voltage supply. This part is admittedly a bit messy since it wasn't planned properly during the design of the prototype.

Take a 4x2 male pin header and create conducting connections between the pins so that the pins in the two 4x1 rows are connected to each other (for the prototype, a small PCB was created for this on which the pin header was placed). Now take the female pin header with wires soldered to it that belongs to patch 6 of the main board. Solder the still open ends of the wires to a 2x1 female pin header and plug it into the 4x2 male pin header so that one row is now connected to GND, the other to VCC when the wires are plugged into the main board. Take the two female pin headers belonging to patch 1 and 3 of the input/output boards and plug them into the 4x2 male pin header so that the connections to GND and VCC are correct.

There is place for one more 2x1 female pin header left on the 4x2 male pin header. This is for the voltage supply of the breadboards. Tear off a bit of the foam on the bottom of the breadboards to lay bare the metal strips for the + and - lines of the breadboard. Solder wires to the strips as shown in the image below.

After doing this, both breadboards should have a pair of wires soldered to a 2x1 female pin header. Plug the two 2x1 female pin headers into a 2x2 male pin header. On the bottom of the 2x2 male pin header, solder the two pins connected to the + lines of the breadboards to a single wire, and the pins connected to the - lines to another wire. Solder these wires to the two poles of the two pole rocker switch. Solder the other contacts of the two poles to two wires and the other ends of the wires to a 2x1 female pin header. Finally, plug this 2x1 female pin header into the 4x2 male pin header so that the + lines of the breadboards are connected to VCC via the rocker switch and the - lines to GND. The rocker switch allows the
user to disconnect the breadboards from the voltage supplied by the Arduino Uno on the main board so that he/she can safely place components on the breadboard.

Lastly, solder one wire of the battery holder directly to the corresponding pole of the cable with barrel connector. Solder the other wire of the battery holder and the corresponding pole of the cable to the single pole rocker switch. This is the power switch for the whole Flowboard (it only has an effect if the Flowboard is powered by batteries. It doesn’t affect the power supply when running on power via USB). Plug the barrel plug into the Arduino Uno.

Software

The Arduino Uno in the Flowboard runs a version of ConfigurableFirmata. ConfigurableFirmata, as the name suggests, is a version of Firmata that lets the user configure the feature set of the protocol relatively easily due to its modular structure. To take the ease of configuration a step further, the Firmata Builder allows to select the desired configuration by clicking through the website. The sketch used in the Flowboard is provided with these instructions (FirmataFlowboard.ino). Apart from the functionality available on Firmata Builder, the customized Firmata version of the Flowboard also adds support for buzzers, WS2812 NeoPixel strips and SSD1306 driven OLEDs.

Before installing the FirmataFlowboard.ino sketch to the Arduino Uno, all the included libraries have to be installed to the Arduino IDE. Install the latest version of the ConfigurableFirmata library by following the instructions in the link. Replace the ConfigurableFirmata.h file in the Arduino/libraries/ConfigurableFirmata/src folder of your Arduino IDE installation with the file provided with these instructions. Install the Adafruit NeoPixel and ssd1306 libraries by following the respective instructions. Lastly, the wrapper libraries for the added functionality provided with these instructions need to be installed. To do so, move the folders FirmataAdafruit_NeoPixel, FirmataBuzzer and FirmataSSD1306 to the Arduino/libraries/ folder of your Arduino IDE installation. You may need to restart the Arduino IDE if it is currently running for the installation to take effect. Finally, load the FirmataFlowboard.ino sketch onto the Arduino Uno. With this, the Arduino Uno is ready to go for the Flowboard and can be plugged onto the main board of the Flowboard.

The last thing remaining is to take the Flowboard Xcode project provided with these instructions and to build it on the iPad. Once a developer team is added in the project editor, the project should be able to build. Don’t forget to turn on Bluetooth on the iPad for the communication with the Flowboard!

Casing

At this point, the Flowboard is in principle ready to go, but the huge amount of wires make it cumbersome to use in this form. For the prototype, a rudimentary laser cut plywood casing was used to house the Flowboard.
The SVG file for reproducing the case is provided with these instructions and depicted in the image below.

![Diagram of the case casing](image)

As the images show, the casing consists of three main plates held together by the side pieces, creating two layers within the case. The lower layer houses the main board and the batteries, while the breadboards and the iPad are placed into the upper layer. The cutouts labeled with letters in the image above have following functions:

(a) Cutout for iPad. The iPad is placed on the middle plate, with the top plate placed above it. The cutout is just the size of the the iPad display.

(b) Cutouts for breadboards. The breadboards are sticked onto the middle plate, these cutouts make them reachable for the user.

(c) Cutouts for the input/output boards. The input/output boards are sticked on top of the top plate over these cutouts that make room for the pin headers and the wires on the bottom of the boards.

(d) Cutouts for wires from the the input/output boards. The wires are funnelled through these cutouts to the main board, which is placed on the bottom plate.

(e) Cutout for battery holder. The battery holder is snapped into this cutout from the bottom so that the user can access the batteries from the outside at the bottom of the Flowboard.

(f) Cutout for power switch that turns voltage supply on the breadboards on and off.

(g) Cutout for power switch that turns the whole Flowboard on and off.

(h) Cutout for USB cable. Align the main board with the Arduino Uno on it so that the USB port on the Uno is reachable through this cutout.
Assemble the casing by starting with the lower layer, then the middle layer and at last add the upper layer to close the device. Just start and go on by trial and error until everything is in place, nothing else happened during the assembly of the prototype you see in the photos above. Two pairs of hands, some tape and a soldering iron to desolder and resolder some wires proved to be useful. If everything is in place and you are sure that you won’t need to open the FlowBoard anymore, you can glue the plywood parts together. Otherwise, use tape to hold everything together. Especially for the top plate it’s useful to be able to reopen it every now and then to access the iPad.