



Monterey Bay Aquarium Research Institute

ESP MicroFluidics Block (MFB)

User's Manual

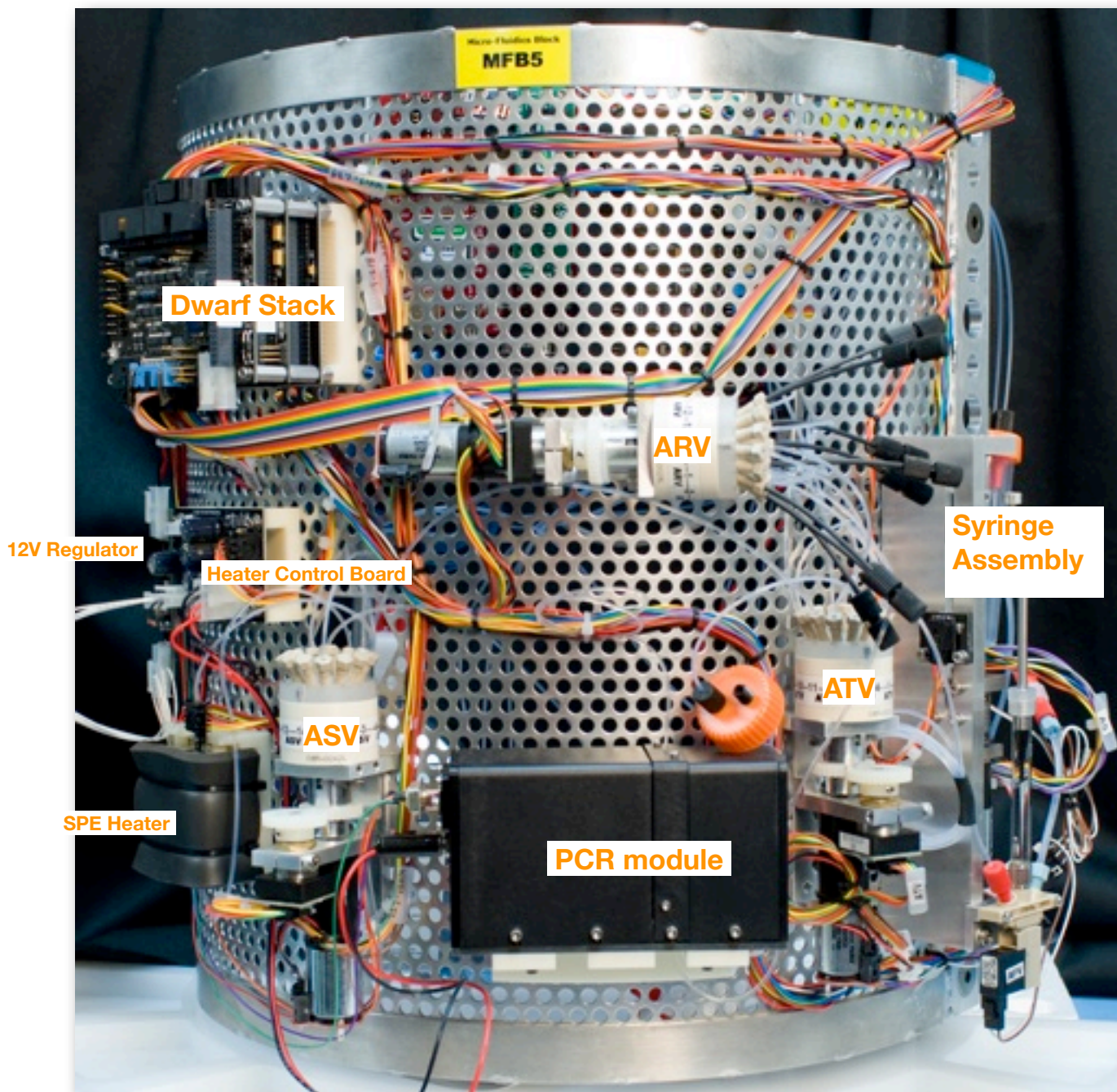
v. 1.0.4

Parts Identification	3
Front Grill-----	3
<i>Analytical Syringe (AS) -----</i>	4
<i>Analytical Fluidics Valve (AFV) -----</i>	4
<i>Analytical Reagent Valve (ARV)-----</i>	5
<i>Analytical Top Valve (ATV)-----</i>	5
<i>Analytical Sample Valve (ASV)-----</i>	5
<i>Solid-Phase Extraction (SPE)-----</i>	6
<i>Heater control board -----</i>	6
<i>Dwarf stack -----</i>	7
<i>12V regulator board -----</i>	7
Back Burner-----	8
<i>ARM Host board -----</i>	9
<i>Dwarf Gateway -----</i>	10
<i>I2C power regulator board-----</i>	11
<i>Analytical Injection Valve (AIV) -----</i>	11
<i>Analytical Waste Valve (AWV) -----</i>	11
Fluidics Pathway-----	12
Electronics Layout-----	14
Power Requirements	14
Battery power supply-----	14

<i>Powersonic PS-12120</i>	14
Battery charger	14
<i>Xenotronic HPX-30</i>	14
Communications	15
Establishing communications	15
<i>Start-up</i>	15
<i>Accessing MFB</i>	17
Running the Instrument	18
<i>Individual commands</i>	18
<i>Syringe Movements</i>	18
<i>Valve Movements</i>	20
<i>SPE Commands</i>	21
<i>PCR Commands</i>	22
<i>Cleaning Commands</i>	23
<i>Configuring MFB</i>	24
<i>ARV Protocols</i>	25
<i>Analytical Syringe (AS) protocols</i>	25
<i>PCR Reaction Building Protocols</i>	26
<i>Reagents for Solid Phase Extraction</i>	27

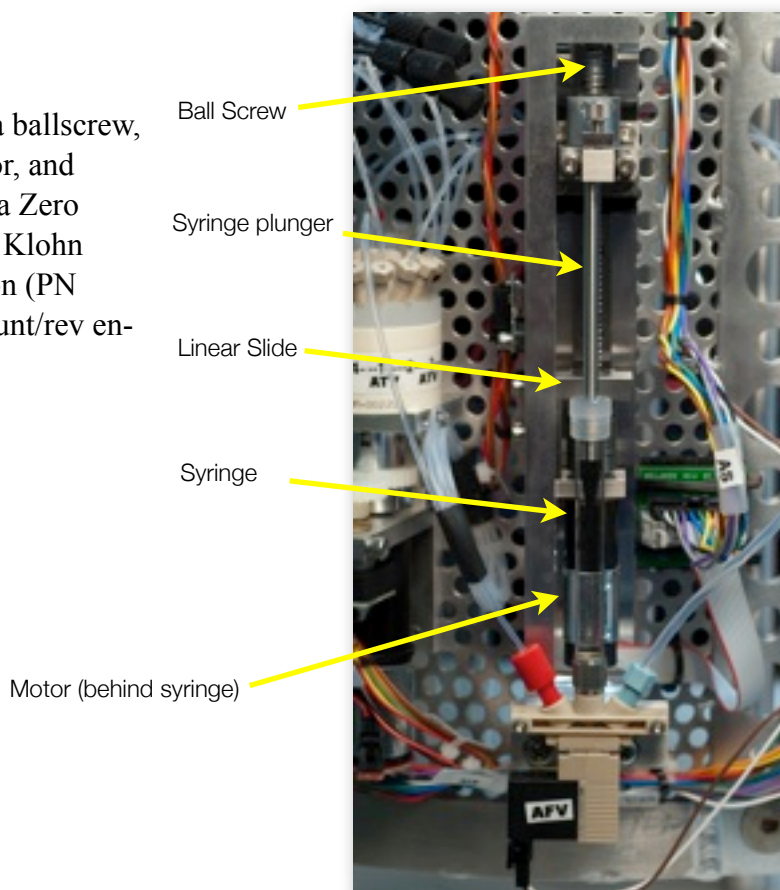
Parts Identification

Front Grill



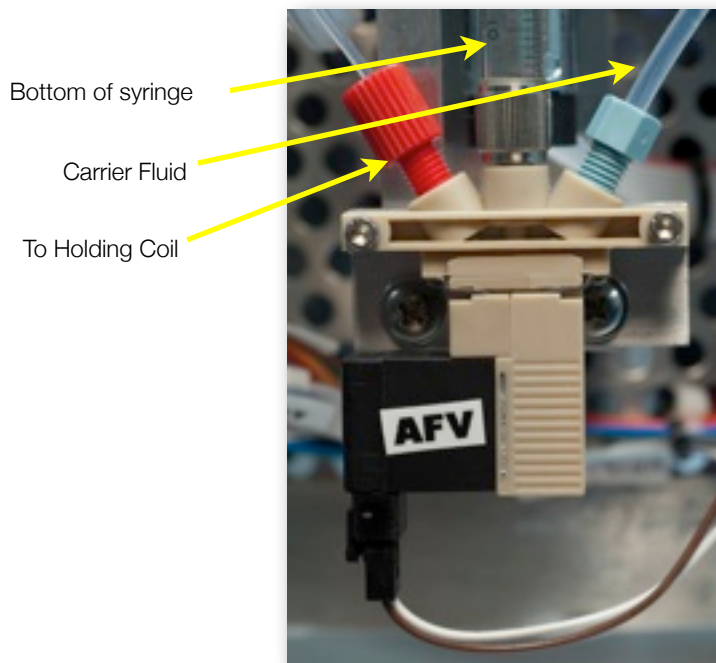
Analytical Syringe (AS)

The syringe assembly consists of a ballscrew, linear slide, coupler, syringe, motor, and home flag sensor. The syringe is a Zero Dead Volume 1.0 ml syringe from Klohn (PN 29557). The motor is a Maxon (PN 348495) 12V motor with a 512 count/rev encoder.



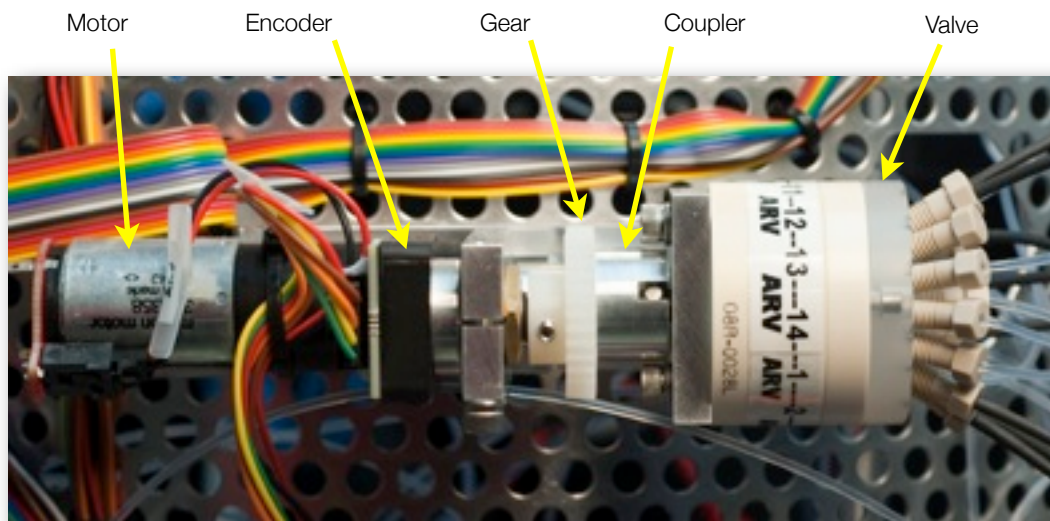
Analytical Fluidics Valve (AFV)

This is a 3-way flipper solenoid valve and manifold located at the bottom of the syringe. One side connects to the carrier fluid, the other to the holding coil, with the syringe being common between them.



Analytical Reagent Valve (ARV)

All rotary valves (either 14-port or 6-port) have the same construction; a motor connects to a geared coupler that connects to the valve. Off the geared coupler is another gear that drives the encoder. These parts are labeled for the ARV below. The ARV supplies the ATV and is mainly involved in supplying the various reagents (primers and master-mixes for PCR). Note heat-shrink tubing covering reagent tubes to protect light sensitive reagents.



Analytical Top Valve (ATV)

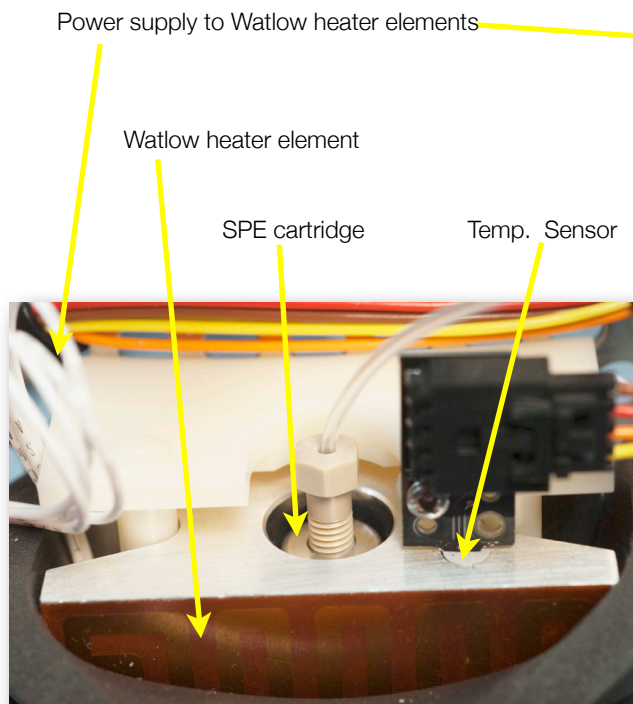
This 14 port stream-select valve is center supplied by the holding coil and operates as the central valve allowing access to the core ESP, reagents on the ARV, and modules through the ASV. The assembly of the ATV is the same as the ARV.

Analytical Sample Valve (ASV)

This 14 port stream-select valve has a special stator that accesses the mixing coil and delivers reagents to the SPE heater and Analytical Module. When performing PCR, it also serves to isolate the PCR reaction in the module during thermocycling and ensures amplified products are moved away from the central ATV valve. See the Fluidics diagram for stator configuration.

Solid-Phase Extraction (SPE)

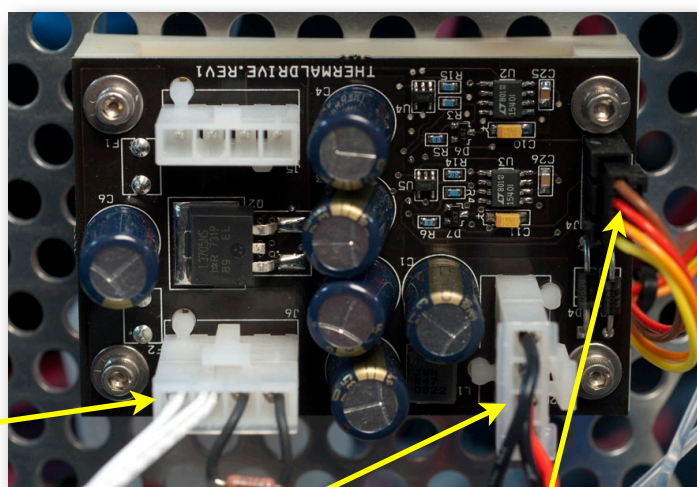
The solid phase extraction heater consists of an insulated aluminum block with two thin-film Watlow heater elements affixed to either side. The center of the block has a column holder that contains replaceable extraction cartridges. Fluid connections to the ASV are made at the top and bottom of the column. There is also a temperature sensor potted into the block that is used for temperature control of the heater.



Heater control board

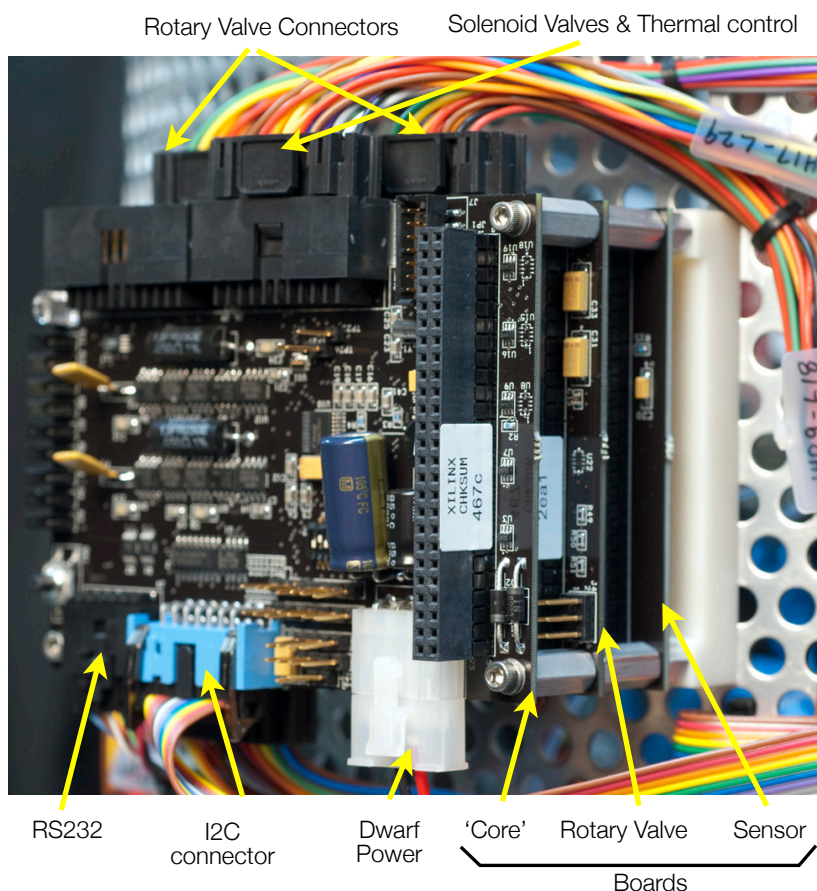
The Heater Control Board receives commands from the Dwarf Stack and supplies the correct amount of power to the SPE heater elements to maintain a desired temperature.

Wiring to Watlow heater elements



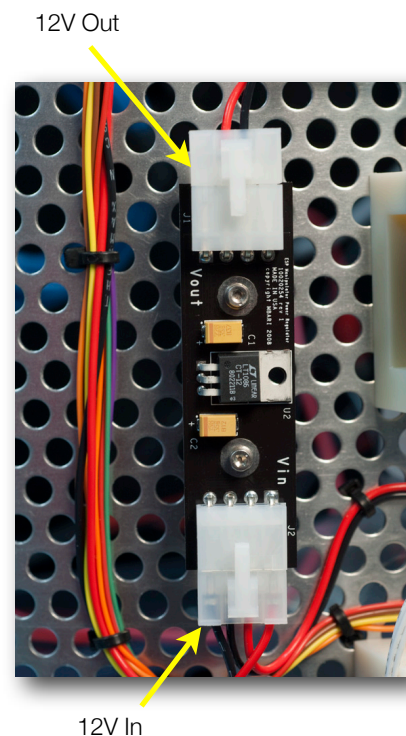
Dwarf stack

The Dwarf Stack consists of a 'Core' board, Rotary valve board and a Sensor board. The core board receives instructions from the ARM host board via the I2C network, and directly controls up to 8 solenoid valves and up to two heaters. The core board also passes RS232 from the Host to an Analytical Module. The Rotary valve board can control up to 4 DC brushed motors and their encoders. The sensor board can receive input of 1-2 temperature sensors and a pressure sensor.



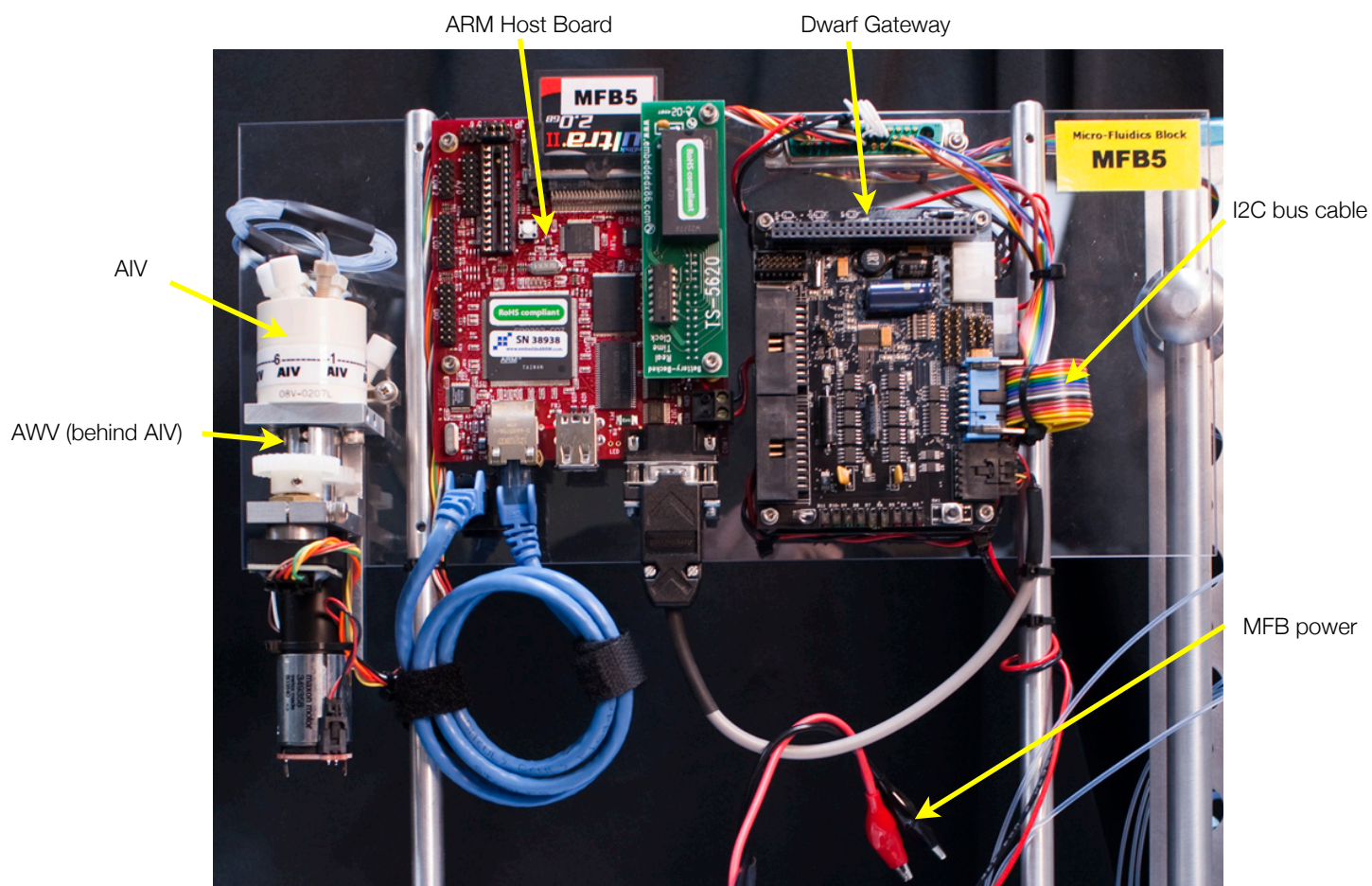
12V regulator board

The 12 Volt Regulator takes raw unregulated power in and supplies a clean 12.0 Volts out to the Dwarf stack. It also passes unregulated power to the Heater control board and to the Analytical Module.



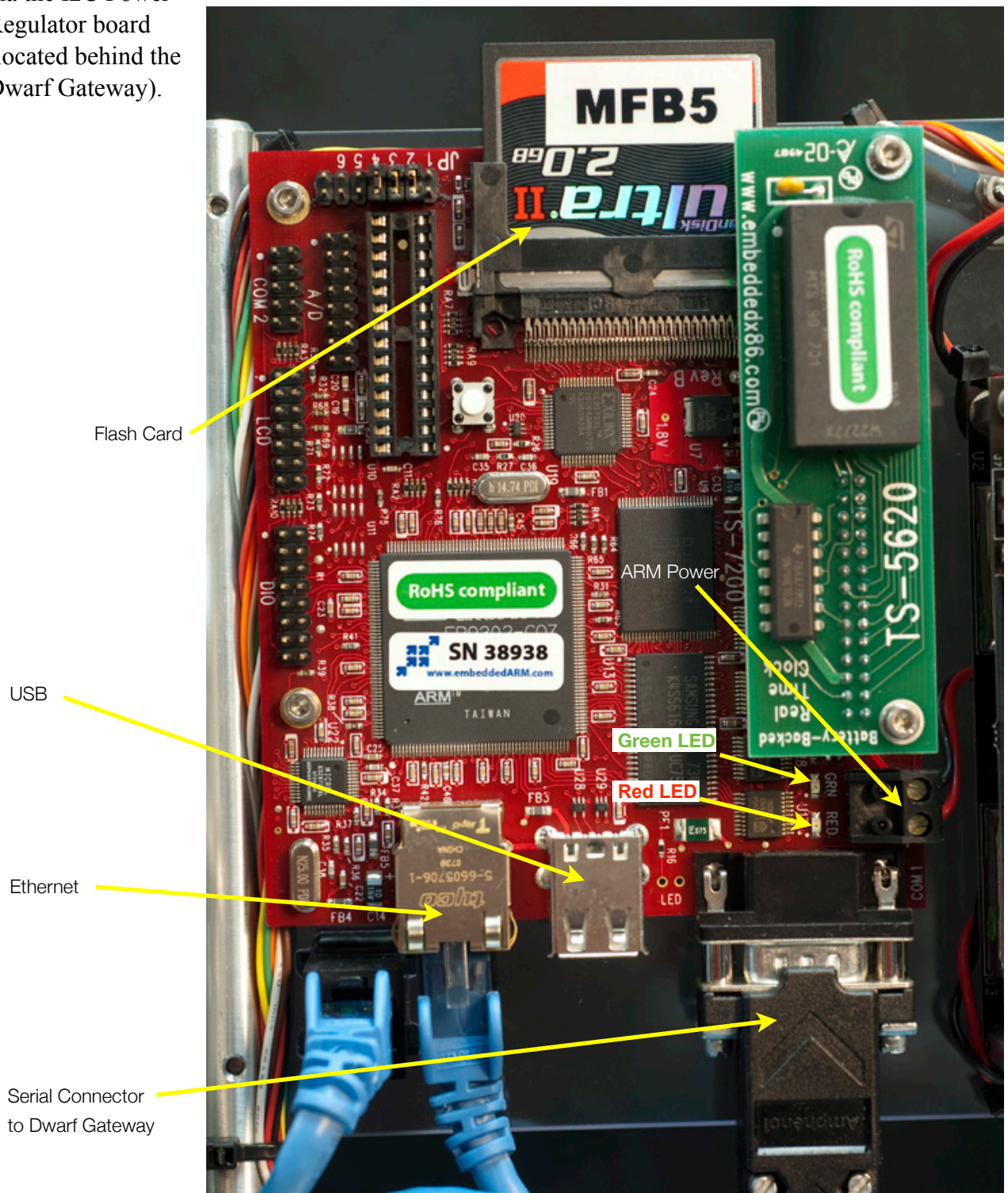
Back Burner

For the MFB to function as a stand-alone device (not connected to an ESP), several boards and valves are included and mounted behind the grill on an acrylic sheet. This is also where power is supplied to the unit (10-16VDC).



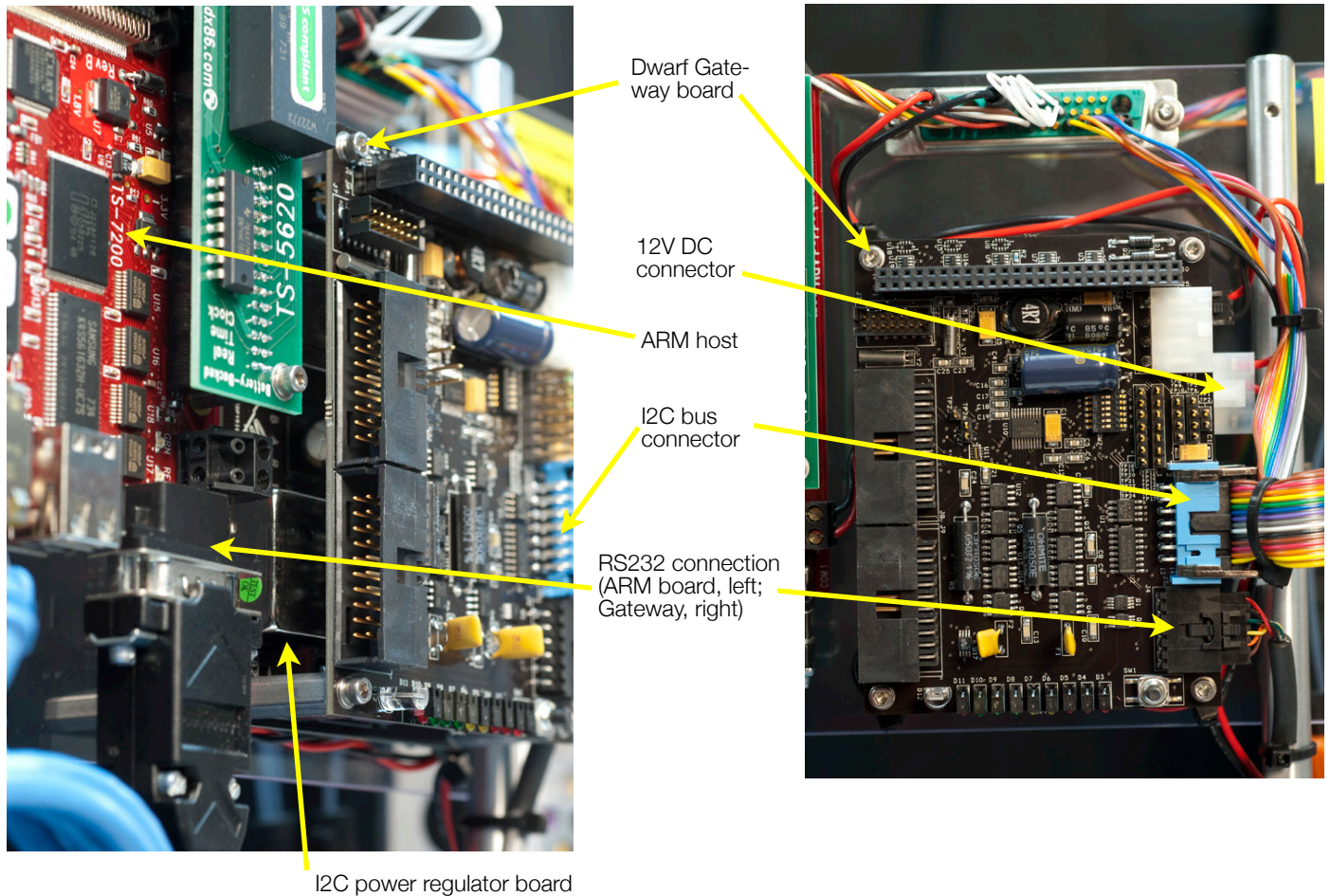
ARM Host board

The ARM host board is the main ‘brain’ of the MFB. It communicates via ethernet or a serial port. The operating system, Linux, lives here, and there is an on-board 2GB Compact Flash for data storage. The board also supports two USB 1.1 peripherals. 5 V DC power is supplied via the I2C Power Regulator board (located behind the Dwarf Gateway).



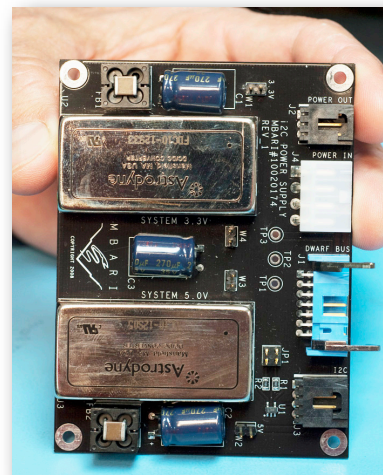
Dwarf Gateway

The gateway is a 'Core' board that has been programmed for use as a gateway to the Dwarf stack. It is connected to the Host board via a RS232 cable, and to the Dwarf stack via the I2C bus.



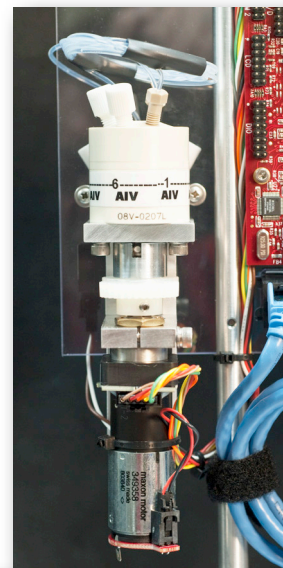
I2C power regulator board

The I2C power regulator board receives 12-16 VDC power from a battery and conditions it for 3.3V and 5V system power needed by the Host board, gateway and the Dwarf stack. It supplies that power via a 14pin IDC connector that is the I2C bus cable. This board is located behind the Dwarf Gateway board.



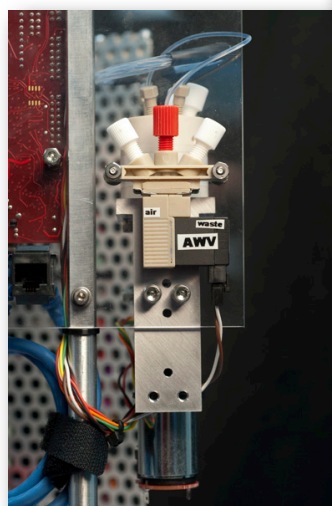
Analytical Injection Valve (AIV)

This 6 port rotary valve is actually a 2-position valve. Position 1 allows manual filling of the lysate coil with sample. Position 2 passes the sample to the MFB.



Analytical Waste Valve (AWV)

Located behind the AIV, this 3 way flipper solenoid valve and manifold provides access to i.) clean air, which ensures an uncontaminated sample is delivered to the MFB, or ii.) waste, in order to decontaminate the injection loop in preparation for the next sample.

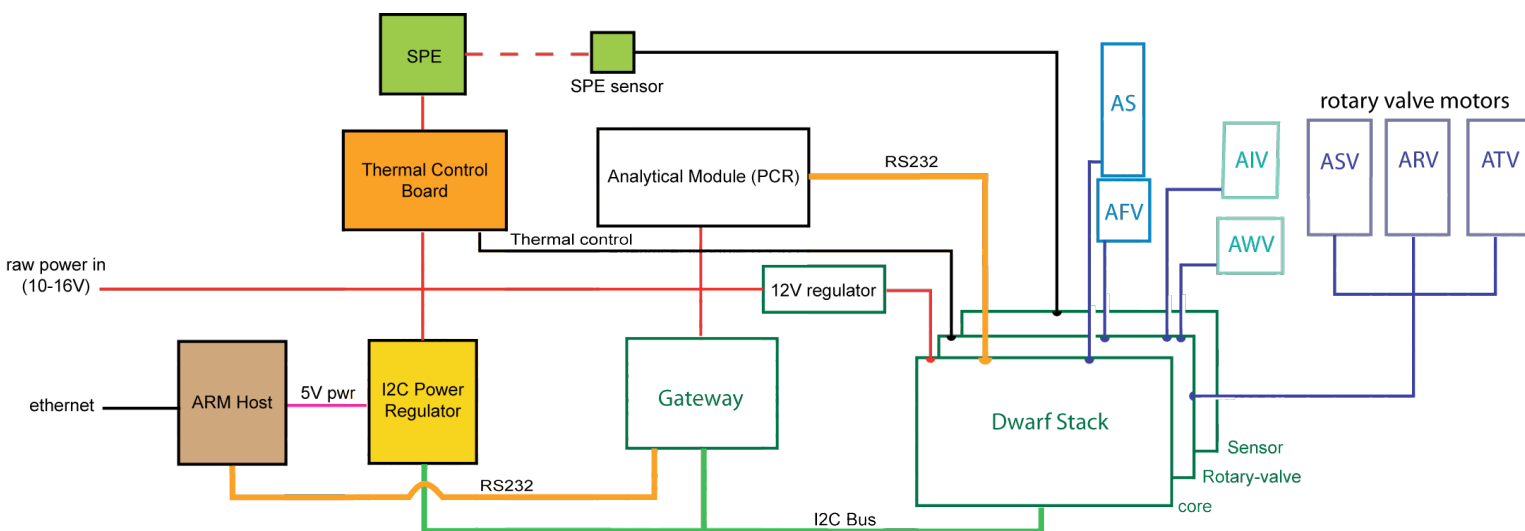


Fluidics Pathway

The schematic on the next page shows valve and syringe arrangement. All valves have a 'center-to-single-port' stator except the ASV and AIV. Reagents on the ARV are left blank, to be filled in by the user according to their needs.



Electronics Layout



Power Requirements

Battery power supply

Powersonic PS-12120

A Powersonic PS-12120 F2 12 Volt, 12 Amp.Hr. battery is used to provide un-interrupted power to the instrument.

Battery charger

Xenotronic HPX-30

A Xenotronic HPX-30 Sealed lead acid battery Charger is used to keep the battery at full capacity. It outputs 12 Volts, and 2 Amps.

Communications

Establishing communications

Start-up

The host computer is normally plugged into a wired Ethernet network via the [RJ-45](#) jack on ARM processor on the back-burner lucite panel. It should integrate itself automatically into any network that supports both [DHCP](#) and [DDNS](#).

If your network supports [DHCP](#), once you have unpacked and visually inspected the MFB for damage, you should plug it into your network via a [Cat 5 Ethernet cable](#) before applying power to the MFB. When power is applied, a pair of tiny red and green light emitting diodes on the component side of the host computer board will blink briefly. One of the lights should remain solidly green if the board is working properly. If the lights continue to flash in a rapid, alternating pattern, the computer is failing to start. In this case, remove power and try again. If the problem persists, call MBARI.

Soon after the boot LEDs finish blinking, you should be able to observe two other blinking LEDs on the RJ-45 jack attached to the host computer board. Note that there are no LEDs on the MFB's Lucite panel. Follow the patch cable from that jack to the host board to see the network status LEDs. At least one of these should be illuminated and you should see irregular blinking corresponding to network activity. If these LEDs are still dark after 20 seconds, there is most likely a problem with the network connection. Check cables and routers. Try plugging a laptop or other computer into the same cable you intended to use for the MFB. If the other computer works, but the MFB shows no network lights, try another cable and/or try plugging directly into the [RJ-45](#) connector on the MFB host computer, bypassing the patch cable to the Lucite panel.

If the network is plugged in when the MFB host is first powered on or reset, it should be ready to contact 30 seconds after you first observe network activity on its LEDs. However, if the network is plugged in after the host computer as started booting, the network integration process may take as long as five minutes. During this process it queries your network router for a unique [IPv4](#) address and requests that it be associated with the unique name assigned to the MFB at the time of its manufacture. This "hostname" will be displayed on a prominent label affixed to the MFB's Lucite panel.

You can test whether the MFB is accessible a number of ways. In all the examples below the MFB's hostname is to be substituted for 'hostname' in the text:

- Point a web browser on your network at the URL: `FTP://hostname`
- Telnet to the host with the command: `telnet hostname`
- Ping the hostname with the command: `ping hostname`

- SSH into the host with the command: `ssh mfb@hostname`

Note that commands are to be entered on the command line of a computer on the same [LAN](#). SSH is available under Mac OSx, Linux and Unix, but may not be available under Microsoft Windows unless it had been installed as a 3rd party add on. The ping command on Linux and Unix will continue testing until aborted (typically with 'Control-C'). If either telnet and/or ssh prompt for login or password you are ready to begin using the instrument.

The MFB host computer supports two networking protocols to log in to it and issue Linux commands. The telnet protocol is not secure, but supported on just about every networked computer. The ssh protocol, or Secure SHell, is supported by all Linux, Unix and Mac OSx computers. Use ssh if your computer supports it.

Telnet always prompts for both a username and a password. SSH infers the user name from parameters input on the command line and prompts only for the password. Here's a typical command dialog for Telnet starting a telnet session: (typed text is in ***bold italics***)

```
> telnet MFB3
Trying 134.89.10.238...
Connected to mfb3.shore.mbari.org.
Escape character is '^]'.

MBARI ESP Embedded Linux 1.7 -- 12/2/08 brent@mbari.org
MFB3 login: mfb
Password: mclane
Using esp2local, ESPhome=/home/mfb/esp2
mfb@MFB3:~$
```

Here's a typical command dialog for starting an ssh session:

```
> ssh mfb@MFB3
The authenticity of host 'mfb3 (134.89.10.238)' can't be established.
RSA key fingerprint is ae:05:da:5f:4a:05:a1:7b:b8:be:2b:64:4a:00:a4:06.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'mfb3,134.89.10.238' (RSA) to the list of known
hosts.
mfb@mfb3's password: mclane
Using esp2local, ESPhome=/home/mfb/esp2
mfb@MFB3:~$
```

Both the Telnet and SSH sessions function identically once started. One may have any reasonable number of either session type opened simultaneously.

If the above commands indicate that the MFB host computer is not reachable, it may be that your network supports [DHCP](#) but does not support [DDNS](#). This means that the MFB can only be identified by its numeric IP address of the form 11.22.33.44. This numeric address is typically assigned by your network router box. Search your router's configuration web pages for information relating to DHCP assignment on your local network. You may find an entry for the MFB host name listed there. If you do, try substituting the IP address your router assigned in place of hostname in the above commands. If that works, you can identify the MFB host by this number. However, beware that it may change whenever the MFB is removed from the network or powered off.

Accessing MFB

To start the esp application that controls the MFB from the host's Linux command prompt:

```
esp@hostname:~$esp
```

Allow 30 seconds for the esp application to start and issue the esp command prompt:

```
@20:48:47.63 <MAIN> Copyright 2008 MBARI
/home/mfb/esp2/mfb/MFB3/configure.rb
$Revision: 1.1 $by $Author: brent $on $Date: 2008/12/06 05:47:00 $
@21:04:10.64 SPE.stop
@21:04:10.70 SPE.configure SPEconfig
@21:04:10.83 AS.coast
@21:04:10.87 AS.configure ASconfig
@21:04:11.05 PCR.pidtemp= 50,0.1,0,300,25,25
@21:04:11.15 PCR.caltemp= 4.803,0.0371,0
@21:04:11.19 All dwarves are running firmware version 2.59
ESPMFB3:001:0>
```

You should see messages indicating that various mechanical components of the MFB are being configured and initialized. If you see messages indicating that parts are "not responding" or "MISSING", execute the shutdown procedure described below and check cabling, etc., then power up, log in and try starting the esp application again.

The esp application is written in the programming language [Ruby](#) and its command prompt is a variant of Interactive Ruby or [irb](#). It supports command auto-completion with the <TAB> key and command history recall using up and down arrow keys.

Ready the instrument - When the MFB is switched on, one must calibrate its Analytical Syringe and other components before using it. The command to do that is:

>**MFB.ready!** (MFB.ready! will home the syringe and flush 1 ml of carrier fluid through the syringe and out the ATV waste port. The MFB is left with the syringe at empty, ATV at waste, ARV between carrier and air1, and ASV at waste1.)

>**MFB.ready?** This returns true if all axis are homed, false if not.

To quit the ESP application and return to the Linux command prompt:

```
ESPMFB3:003:0> quit
mfb@MFB3$
```

--or--

```
ESPMFB3:003:0>exit
```

Running the Instrument

LeTtEr-CaSe Matters!!

Individual commands

Syringe Movements

Before moving the syringe, always check the position of the valves!

- To initialize the home sensor
>**AS.home.to :home**
- **Configuration names** for syringe speeds:
We sample at 64 tics/sec, thus 1 tic = 0.015625 sec

ASslow0: maxSpeed = 200 counts/tic = 10 ul/s

ASslow1: maxSpeed = 100 counts/tic = 5 ul/s

--ASslow1 is used majority of time to get & deliver reagents

ASslow2: maxSpeed = 20 counts/tic = 1ul/s

--ASslow2 is used to maintain the integrity of reactions separated from the carrier fluid by air bubbles or AS moves <10µl. If ASslow2 is used the syringe must be above the 0 or empty position. The stiction of the syringe at empty (or 0) does not allow metering of accurate volumes.

ASconfig: maxSpeed = 350 counts/tick = 17.5 ul/s

ASnorm: maxSpeed = 400 counts/tic = 20 ul/s

ASfast: maxSpeed = 750 counts/tic = 37.5 ul/s

- To fill syringe (default speed ASnorm)
>**AS.fill**
- To empty syringe (default speed ASnorm)
>**AS.empty**
- To move at a 'custom' speed (in ul/s)
>**AS.config.maxSpeed=17**
>**AS.configure**
- To change syringe speed on the fly
>**AS.to position, configurationname** (e.g., **AS.to .5, ASfast**)
>**AS.pull volume, configurationname** (e.g., **AS.pull 0.3, ASslow1**)

For AS.fill and AS.empty, no comma is needed

>**AS.fill ASfast**

- To move to a absolute position on the syringe (e.g., 0.6ml)
>**AS.to volume** (e.g., **AS.to 0.6**)
- To push a relative volume of fluid
>**AS.push volume** (in mL) (e.g., **AS.push 0.01**)
- To pull a relative volume of fluid
>**AS.pull volume** (in mL) (e.g., **AS.pull 0.01**)
- To stop the syringe
Either >**control-C** (if the last command was a syringe move)
or >**AS.stop** (if it is doing something else also)
- To release the syringe motor from holding a position
>**AS.coast**
- To query the raw encoder counts for calibration purposes
>**AS.rawPosition**
- To change syringe speed (use configuration names above)
> **AS.reconfigure configurationname** (e.g., **AS.reconfigure ASslow1**)
- To fill the syringe full of carrier solution (1 ml)

>fillSyringeFull

- To fill the syringe half full of carrier solution (0.5ml)

>fillSyringeHalf

- To pull a volume a bleach from the ATV into the syringe.

>bleach(volume)

(Note: this protocol pushes 15 ul of air back into the valve port)

- To safely pull a volume of ethanol

>EtOHwash(volume)

(Note: this protocol pushes 15 ul of air back into the valve port)

Valve Movements

There are six valves: ATV, ASV, ARV, AIV, AFV, and AWV

- To find the current position of any valve
>valvename (e.g., **ATV**)
- To find out the names of the valve ports
>valvename.legend (the result is a list of encoderValues mapped to port numbers and names)
- To move to a port on any valve
>valvename.to portnumber (e.g., **ATV.to 1**) or
>valvename.to :portname (e.g., **ATV.to :PCRwaste**)
>valvename.portname (e.g., **ATV.PCRwaste**)
- To block off flow by moving valve exactly between ports x and y:
>valvename.dialBetween portnumber, nextportnumber (e.g., **ATV.dialBetween 1,2** or **ATV.dialBetween :carrier, :air1**)
- To move the valve in a specific direction:
>valvename.to portnumber,:direction (e.g., **ATV.to 8,:up**) (e.g., **ATV.to 8,:down**)
- Another way to move valves, crossing or avoiding certain ports:
>valvename.to portnumber, :via => :portname
:avoiding => portname
- To get the encoder position of the Valve
>valvename.rawPosition (e.g., **ATV.rawPosition**)

- To rotate the valve by x
 >*valvename.advance x*
 >*valvename.retard x*

(Note: x may be signed and/or fractional, thus >**valvename.retard -1.5** is equivalent to >**valvename.advance 1.5**)

SPE Commands

Commands for working with the Solid-Phase Extraction (SPE) component of the MFB:

- To determine the temperature on the SPE heater
 >*SPE*
- To access the raw data of the SPE heater
 >*SPE.status*
- Set the SPE heater for a specified time to a temperature
 >*SPE.hold "hour:min:sec", temp*
- Turn off the SPE heater
 >*SPE.stop*
- To prepare a new SPE column for extraction
 >*columnBurnIn*
- To run solid phase extraction from the AIV:
 >*runSPE* (this defaults to the injection loop on the AIV)

At the end of this protocol, the extracted nucleic acid will be primed to the ATV.11, bleach will remain parked in the SPE extraction column, and ATV-AIV-lysate coil will contain (from the ATV valve), air, water, air, bleach.

- To run solid phase extraction from sample line on the ATV.

You must first prime the lysate to the ATV valve. Prior to priming, dip the end of the tube in 1 ml of 20% (v/v) Chlorox bleach (unscented), followed by a dip in two tubes of 1 ml clean water. Place the tube in the modified lysate.

```
>ATV.to :valveposition
>AS.pull 0.15, ASslow0
>ATV.to :waste
>AS.empty
>runSPE :valveposition
```

At the end of the protocol, the extracted nucleic acid will be primed to the ATV, bleach will remain parked in the SPE extraction column, and the line that the lysate was primed to will contain (from the valve), air, water, air, bleach.

- To recover the lysate from ATV.13 (:extSrc) after the extraction
>deconEluateRecoveryLine :extSrc
>recover (volume)
 Either the whole extract can be recovered or multiple aliquots of the sample. The extract is recovered through ATV.13 (:extSrc) into 1.5 ml epi tubes.
- To decontaminate the SPE module and ready it for the next sample
>deconSPE
 Note: deconSPE defaults to decontaminating the AIV line.
- If lysate was loaded via an external port on the ATV.
>deconSPE :portname

At the end of the decontamination, the following have been aired out: ATV-ASV line, column, mixing coil, and lysate delivery line.

PCR Commands

- To turn the PCR fan on/off
>PCR.fan = :on >PCR.fan = :off
- To query the PCR.state
>PCR.status
- To set PCR to a temperature
>PCR.temp = tempdesired
- Returns the current PCR program
>PCR.program
- Loads a new PCR program
>PCR.program :programname
 Programs are found/created in the MFB/MFB#/configure.rb file
- To run the PCR program
>PCR.run

- To visualize PCR data

>PCR.sample

Once a new run is initialized the data from the previous run is lost.

- Stopping PCR

>PCR.abort (stops any PCR run in progress)

>PCR.off (stops any PCR run in progress and turns off PCR heating element)

Cleaning Commands

- To decon the AIV Lysate line (done automatically in deconSPE or fullDecon)
>deconLysateSA (if MFB is in Stand-Alone mode)
>deconLysateINT (if MFB is mated to the core ESP [i.e., INTEgrated])
- To decon after Solid Phase Extraction
>deconSPE (defaults to cleaning AIV injection loop)
>deconSPE :lysateSource (deconSPE includes the decontamination of the lysate delivery line)
- To decontaminate between PCR runs
>pcrDecon
- To decontaminate after running solid phase extraction and PCR
>fullDecon (defaults to AIV injection loop)
>fullDecon :portName
- To clean the Eluate recovery line or lysate line on ATV
>deconEluateRecoveryLine :portName
- To clean the PCR line when the tubing has been replaced or contaminated (This is a more thorough cleaning that concentrated on the PCR line)
>cleanPCRline (rinses with water followed by pcrDecon)
>waterClean (pcrDecon followed by water washes running ;bleachProfile)

Configuring MFB

Configuration of the MFB is handled in 3 separate files:

```
preconfig.rb
postconfig.rb
configure.rb.
```

The pre- and post- configuration files are found in esp2/mfb. Configure.rb can be found in esp2/mfb/<hostname>

The [preconfig.rb](#) and [postconfig.rb](#) are conditions that apply to all MFBs.

The [configure.rb](#) contains specific configuration information for each individual instrument.

[preconfig.rb](#) contains configuration information for the type of valves on the MFB.

[postconfig.rb](#) identifies each valve name and type and provides the default configuration names for the valve ports on the MFB.

[configure.rb](#) is located in esp2/mfb/MFB#. Machine specific configurations are located here. These include parameters for solid phase extraction, building and positioning the per reaction, and decontamination found in protocol scripts. It also contains calibration information for the SPE heater, analytical syringe, PCR module. The PCR reaction types and their thermocycle conditions are also found here. Within this file you can reconfigure the valve names on the ports.

Below is the current set-up for running PCR (default found in postconfig.rb)

```
unless ARV.positions > 0
ARV.with AxisMap.new(Valve::FourteenPosition)
ARV.rename :air1=>1, :PP1=>2, :MM1=>3, :PP2=>4, :air2=>5, :PP3=>6, :MM2=>7,
:PP4=>8, :air3=>9, :PP5=>10, :MM3=>11, :waste=>13, :carrierSolution=>14
ARV.alias :carrier=>:carrierSolution
End
```

Copy and move to configure.rb for appropriate MFB, then update the port names. Note that ARV.2 has two names, Reagent1 and HybSol. Multiple names for a single port can also be made using aliases (see ARV.alias :carrier=>:carrierSolution).

```
unless ARV.positions > 0
ARV.with AxisMap.new(Valve::FourteenPosition)
```

```

ARV.rename :air1=>1, [:Reagent1, :hybSol=>2], :Reagent2=>3, :Reagent3=>4, :Reagent4=>5, :Reagent6=>6, :Reagent7=>7, Reagent=>8, :Reagent9=>9, :Reagent10=>10, :Reagent11=>11, :waste=>13, :carrierSolution=>14
ARV.alias :carrier=>:carrierSolution
End

```

ARV Protocols

Cleaning pigtails for PCR reagents

To bleach and rinse the lines which connect to reagent coils on the ARV. You will need two Stedim bags, a 50ml bag filled with 20% bleach and a 150ml Stedim bag with clean water, and a short piece of tubing to attached the bag to the port.

Hook up a bag of 20% bleach to the appropriate port.

```

>ATV.to :ARV
>ARV.to :portname
>AS.to 0.5, ASslow0

```

Flip the bag upside down and remove bleach in the neck of the Stedim bag.

```

>AS.fill

```

This should air out the pigtail and ready it to receive the rinse.

Hook up a bag of clean water to the pigtail.

```

>cleanPigTail (port)

```

Once finished hook up a 0.22um GV Sterevix filter to line.

```

>ATV.to :ARV
>AS.to 0.3, ASslow1
>ATV.to :waste
>AS.empty

```

The pigtail line should be aired out.

Analytical Syringe (AS) protocols

Removing Air Bubble from Syringe

Fill a 60cc syringe with a short line of hose (end fitting should screw into AFV valve) with 25ml of water and evacuate all air. Remove the holding coil from the AFV valve. Hook up a short line to the 60cc syringe filled with water.

```

>fillSyringeHalf; ATV.to :waste; AS.empty; fillSyringeHalf

```

(Commands separated by semicolons allow you to string along commands)

As the syringe is emptying, pull on the 60cc syringe to evacuate all contents of the AS syringe (create a vacuum in the barrel of the syringe) and hold it until the AS has emptied and switched back to the carrier fluid on the AFV valve. When the valve flips to the carrier it

should fill with water and then fill to half. If the air bubble has not disappeared, repeat the above line. Once the bubble has disappeared, reattach the holding coil to the ATV and

>3.times {fillSyringeFull; ATV.to :waste; AS.empty ASfast}

PCR Reaction Building Protocols

These commands refer to building a reaction 'slug' trapped between two small air bubbles from reagents on the ARV and ATV. The slug is then moved into position for thermocycling and analysis via the PCR module. Data stored in *.pcr files is comma delineated and can be opened in Microsoft Excel. PCR profiles and locations of the primers and probes are found in mfb/MFB#/configure.rb.

To build a slug and position it in the PCR module:

>pcrSlug(template, mastermix, primerprobe, endAir)

e.g. pcrSlug (ASV, :MM1, :PP2, :air2)

To build a slug, position it, thermocycle and create a file with data

>pcrMission (template, profile)

If the slug is already positioned in the module,

>pcrMission (nil, profile)

(Runs the thermocycling protocols and save the data to a file.)

Alternatively,

>PCR.program :programname

>PCR.run

>PCR.sample

(note data displayed on screen only)

The above protocols require decontamination between reactions.

>pcrDecon

To run combinations of solid phase extraction, multiple PCR profiles and templates with the appropriate decontamination protocol between, use the following variations of the shortmfb command. The commands incorporate the pcrMission, pcrSlug, and SPE protocols. The basic structure of the command is below followed by examples of how it works. A file containing the PCR data is saved for each reaction.

>shortmfb ([profiles], [templates]) {lysatesource}.finish

e.g., ***shortmfb ([IPC, SAR11], [:elutionWater]) {}.finish***

Builds a reaction for the IPC using elution water as the template, positions it, themocycles, stores the data, runs a pcrDecon. Then repeats the above for a reaction with SAR11 primers.

e.g. *shortmfb ([IPC, SAR11], [:ASV]) {:extSrc}.finish*

Does a solid phase extraction from lysate primed to the ATV valve at position ATV.extSrc, followed by a PCR for the IPC using the eluted DNA (primed to the ATV from the ASV), does a pcrDecon, then a PCR for SAR11 from the eluted DNA, then a fullDecon.

e.g. *shortmfb ([IPC, SAR11], [:ASV, :elutionWater]) {:extSrc}.finish*

Does a solid phase extraction from lysate primed to the ATV valve at position ATV.extSrc, followed by a PCR for the IPC using the eluted DNA (primed to the ATV from the ASV), does a pcrDecon, then a PCR for SAR11 from the eluted DNA, then a fullDecon. Then, does an IPC PCR reaction with elution water as the template, runs a pcrDecon, then does a SAR11 PCR reaction with elution water as the template followed by a pcrDecon.

Reagents for Solid Phase Extraction

Reagents are located on the valve positions specified:

20% Clorox Bleach (ATV.1)

Clorox Brand Bleach unscented diluted in water (Sigma #W4502)
0.2µm filtered

Elution Water (ATV.3) and Carrier (ARV.14 and AFV)

Water (Sigma #W4502)

ESTE Buffer 50 ml (EtOHwash, ATV.6)

13.3 ml Water (Sigma #W4502)
0.2 ml 0.5M EDTA (Sigma #E7889)
0.5 ml 1.0M Tris-pH 7.8 (Sigma #T2569)
1.0 ml 4M NaCl (Sigma #S5150)
35ml 200 proof Ethanol

0.1µm filter into Stedim Bag.

Solid Phase Extraction Dilution Buffer (SPEdil, on core ESP PRV)

100µl 3M NaAcetate pH 5.2 (Sigma # S7899)
800µl 200 proof Ethanol