Version date: 2018-10-04
Responsible: Jani Tammi
Document name: Bias Board Evaluation Software Specification

Foresail-1

Bias Board Evaluation Software Specification

University of Turku

Department of Future Technologies

Authors

Date	Name	Worked with parts
2018-09-18	**JTa**	**Initial specification**
2018-09-30	**JTa**	**Updates towards final version**
2018-10-04	**JTa**	**Version 1.0 finalization**

Approved by

_____

Table of Contents

1 General

The bias board designed for Foresail-1 shall be evaluated under radiation and vacuum conditions. An evaluation PCB version is created for that purpose that includes a MCU for testing purposes (which will be removed from the final version of the board). It features a RS-232 port and implements an evaluation and testing protocol.

This document describes the functional requirements of a program which will help the evaluation and testing work.

1.1 Scope

This document is limited to functional specifications of bias board evaluation software and details that describe use cases and the operating environment that are relevant to the specifications.

1.2 Reference documents

PATE-ForeSail-UTU-Internal/Bias board/BiasProtocol, 2018-09-25

1.3 Abbreviations and acronyms

2 Conditions and Requirements

This chapter describes the environment and the usage in which the software will be utilized, limited to those specifics that are thought to have an impact to these specifications.

2.1 Platform

This software will be created specifically for Philipp's laptop, which runs a Linux OS. Solution will be created using Python 3.5 or newer. User interface (details specified in Chapter 4) shall be implemented as an ASCII interface. Only external dependency will be pySerial version 3.3 or newer.

Prerequisites to run the software

- Python 3.5 or newer
- pySerial 3.3 or newer

2.2 Interfaces

This software will use one RS-232 USB adapter, accessible via `/dev/ttyUSB<n>`. Parameters are 115200,8,N,1 (at the time of writing), but *the "BiasProtocol" document holds authority* over these values and it should be referred to.

Human interaction shall be implemented as an ASCII based UI, which will be described in Chapter 4.

2.3 Environmental Conditions

No special considerations are necessary for this solution.

- Ambient temperatures fall roughly within room temperatures.
  Temperatures are not expected to exceed 35'C or go below 5'C.
- Humidity condensation is not expected to occur.
- Conductive particles deposits will not be an issue.
  Like in workshops that may have metal particles in the air.
- The laptop running the software will not be subject to radiation. It may occupy the testing room (where humans cannot stay during the tests), but will not experience any significant increase in radiation that would need to be given consideration.

3 Functional Specifications

This chapter describes the general functional specifications for the software, based on consultation by Philipp Oleynik in 18th of September, 2018.

3.1 Control

This chapter shall contain all functional specifications related to PATE bias board interfacing.

- **Feature; RS-232 interface**
  USB RS-232 adapter provided by Philipp, accessible as `/dev/ttyUSB<n>`.
  Connection parameters are 115200,8,N,1 (unless otherwise dictated by the BiasProtocol document).
- **Implements Bias Board evaluation interface specific protocol.**
  As specified in document "PATE-ForeSail-UTU-Internal/Bias board/BiasProtocol".
- **Value retrieval at set interval**
  By default, solution will automatically poll, save and display all values at 10 second intervals (maximum skew of 100ms is tolerable). Interval shall be user definable, but never less than 1 second. No fractions are accepted.
- **Ability to set PWM[1-4] values**
  When commanded via UI, the solution will issue the corresponding command(s) to set the bias board PWM values (all interactions will follow the available protocol fully - in this case, return messages are inspected for reported errors).
- **Issue calibration -command, twice**
  The command ("`RRR`") will be issued twice when the operator begins the testing.

3.2 Data

- **Save data in CSV dialect specific to Philipp's OS/tools**
  This has been requested to be the US (aka. 'excel') dialect (comma separated, period as decimal separator, minimal quotation).
- **Realtime Timestamps to all logging**
  Philipp's preference is to see realtime timestamps instead of relative timestamps. These will be used in UI and CSV. UI will display only time portion, but the CSV will contain full date and time information for each measurement row.
- **Log bias board values (CSV)**
  All voltages and currents are shown in the UI and saved in to the CSV (in proper dialect).
- **Log commands (CSV)**
  There will be an additional column in the CSV which will record the commands having been issued between the measurements. This will be a concatenated list of commands and replies.
- **Create and maintain metadata on testing sessions (CSV)**
  Metadata/header will contain:
  - Start date and time
  - Firmware version of the Bias Board (as received by '`VERS?`' command)
  - Label/name string (maximum of 80 characters)
    Free form string provided by the operator. Can be left empty.

3.2.1 Data Specification

This section does not concern itself with logging (CSV) or displaying. Data is described as made available by the bias board. Units, data/reply format and typical range for each data item is given. NOTE: typical range is merely informative and the software will not restrict values based on it.

Individual values

1. Total bias generators supply current (`MEAS06?<LF>`)
   mA, `dd mA<LF>OK<LF>`, {00.0 .. 99.9}
2. Bias voltage Tube 1 Detector 1 (`MEAS00?<LF>`)
   V, `ddd V<LF>OK<LF>`, {000 .. 120+}
3. Bias voltage Tube 1 Detector 2 (`MEAS01?<LF>`)
   V, `ddd V<LF>OK<LF>`, {000 .. 120+}
4. Bias voltage Tube 2 Detector 1 (`MEAS02?<LF>`)
   V, `ddd V<LF>OK<LF>`, {000 .. 120+}
5. Bias voltage Tube 2 Detector 2 (`MEAS03?<LF>`)
   V, `ddd V<LF>OK<LF>`, {000 .. 120+}
6. Bias generator supply current Tube 1 Detector 1 (`MEAS10?<LF>`)
   mA, `dd mA<LF>OK<LF>`, {00.0 .. 99.9}
7. Bias generator supply current Tube 1 Detector 2 (`MEAS11?<LF>`)
   mA, `dd.d mA<LF>OK<LF>`, {00.0 .. 99.9}
8. Bias generator supply current Tube 2 Detector 1 (`MEAS12?<LF>`)
   mA, `dd.d mA<LF>OK<LF>`, {00.0 .. 99.9}
9. Bias generator supply current Tube 2 Detector 2 (`MEAS13?<LF>`)
   mA, `dd.d mA<LF>OK<LF>`, {00.0 .. 99.9}
10. Radiation sensing MOSFET 1 drain voltage (`MEAS04?<LF>`)
    mV, `dddd mV<LF>OK<LF>`, {0000 .. 3300}
11. Radiation sensing MOSFET 2 drain voltage (`MEAS05?<LF>`)
    mV, `dddd mV<LF>OK<LF>`, {0000 .. 3300}
12. Analog Supply voltage (`MEAS18?<LF>`)
    mV, `dddd mV<LF>OK<LF>`, {0000 .. 3300}
13. Supply voltage converter temperature (`MEAS14?<LF>`)
    C, `dd C<LF>OK<LF>`, {00 .. 99}
14. Bias (high voltage) converter temperature (`MEAS16?<LF>`)
    C, `dd C<LF>OK<LF>`, {00 .. 99}

Consolidated values (used by the software, instead of individual value retrieval)

```
MEAS99?<LF>
uuu uuu uuu uuu uuuu uuuu -dd 0 0 0 -dd -dd -dd -dd uu -dd uu
-dd uuuu<LF>OK<LF>
```

Each space separated value corresponds to it's index position, based on the above list of individual value retrieval commands. For example, index zero value is the same as can be retrieved with `MEAS00?` Command, but without the (mA) unit descriptor.

3.3 External Interface to Data

There will be no external interfaces to data. Saved CSV files can be used as needed after testing is completed.

3.4 Interfaces

This solution will operate on only one interface; USB RS-232 serial. The physical USB to RS-232 adapter is provided by Philipp.

3.5 Exported Data Specification

The application will write a CSV file for each time it is used to test a bias board.

- CSV file dialect will be US (aka. 'excel'), specific to Philipp's laptop OS and tools.
- Filename will always be "`YYYY-MM-DD HH.MM.SS.csv`" where the date and time are taken from the instant the testing is started.
- CSV file is written with meta-header that contains the following:
    1. Start date and time; "`Started;2018-09-18 17.44.20`"
    2. Label string (queried during start-up): "`Label;<freeform string up to 80 characters>`"
    3. Version and firmware build date of the bias board; "`Firmware;<string returned by VERS? command>`"
- CSV file is written with column header row (16 columns):
    1. "`Datetime`"
    2. "`Total bias generators supply current`" (`MEAS06?`)
    3. "`Bias voltage Tube 1 Detector 1`" (`MEAS00?`)
    4. "`Bias voltage Tube 1 Detector 2`" (`MEAS01?`)
    5. "`Bias voltage Tube 2 Detector 1`" (`MEAS02?`)
    6. "`Bias voltage Tube 2 Detector 2`" (`MEAS03?`)
    7. "`Bias generator supply current Tube 1 Detector 1`" (`MEAS10?`)
    8. "`Bias generator supply current Tube 1 Detector 2`" (`MEAS11?`)
    9. "`Bias generator supply current Tube 2 Detector 1`" (`MEAS12?`)
    10. "`Bias generator supply current Tube 2 Detector 2`" (`MEAS13?`)
    11. "`Radiation sensing MOSFET 1 drain voltage`" (`MEAS04?`)
    12. "`Radiation sensing MOSFET 2 drain voltage`" (`MEAS05?`)
    13. "`Analog Supply voltage`" (`MEAS18?`)
    14. "`Supply voltage converter temperature`" (`MEAS14?`)
    15. "`Bias (high voltage) converter temperature`" (`MEAS16?`)
    16. "`Commands`" (commands that were executed after the last measurement)
- CSV file will be appended with each measurement data row (including concatenated list of commands that were issued after the last measurement).

4 User Interface

This chapter describes the curses based UI and interactions in detail.

4.1 Commandline options

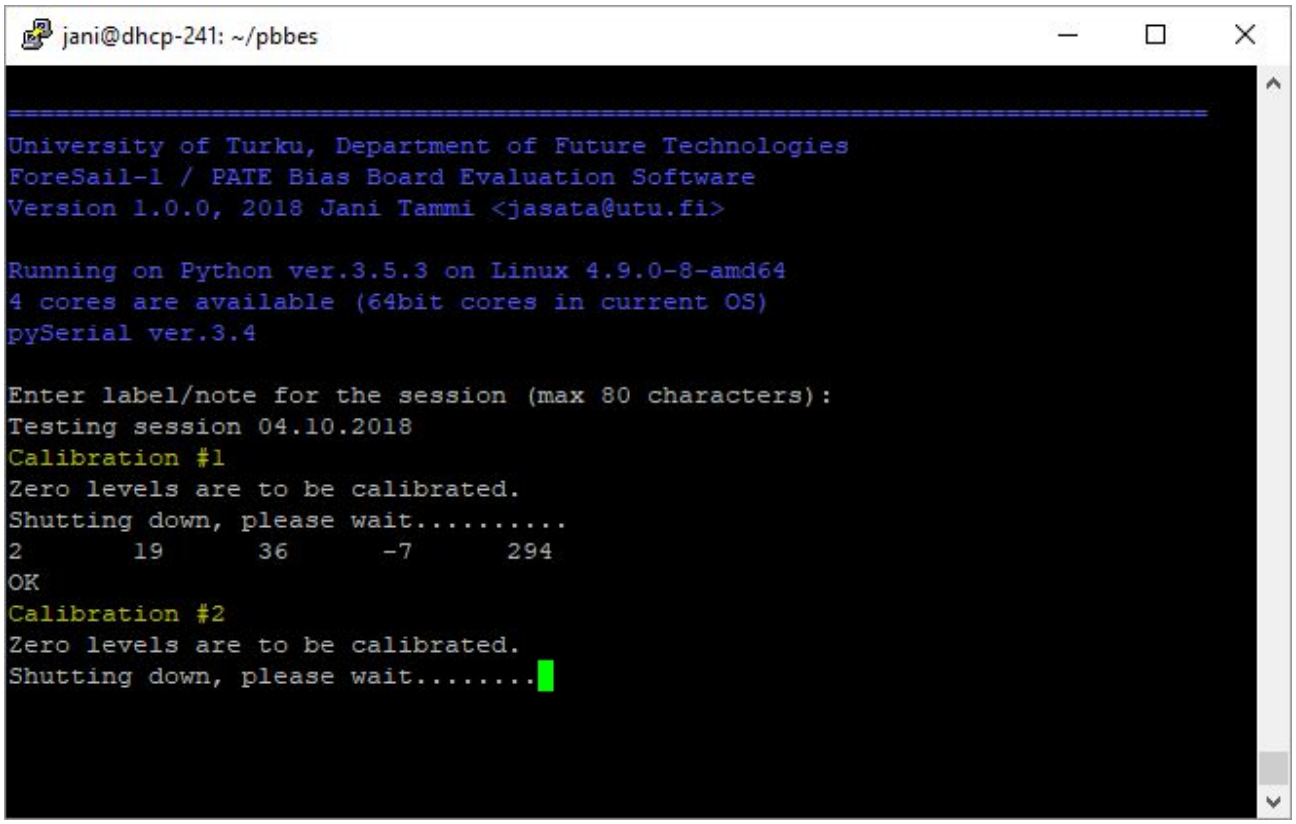
Run parameters that have default values can be given other values by using commandline options. To list available commandline options, '`-h`' or '`--help`' can be specified.

- '`-d`' / '`--device`' to specify serial device.
  Example: `./main.py -d /dev/ttyUSB1`
- '`-l`' / '`--log`' to adjust run time logging level.
  Example: `./main.py -l WARNING`
- '`-i`' / '`--interval`' to specify measurement interval in seconds.
  Example: `./main.py -i 5`

4.2 Start-up Screen

Application will start with a start-up screen which will feature the following:

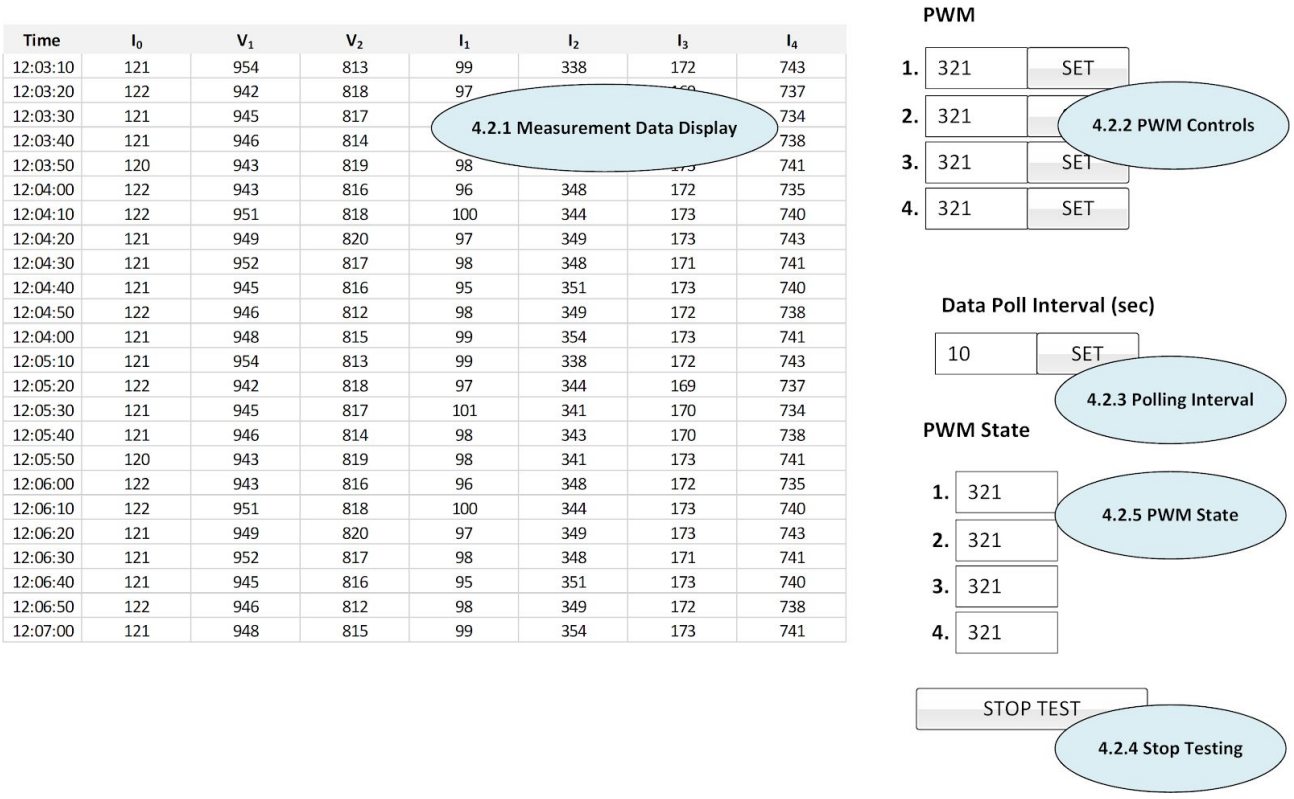
- Allows the operator to enter an optional label for the testing session, which if given, will be written into the CSV header. Field will be maximum of 80 characters.
- Tests serial port connectivity and reports a problem, if any.
- Calibrates the Bias Board by issuing '`RRR`' -command twice (as requested by Philipp) while displaying the output.



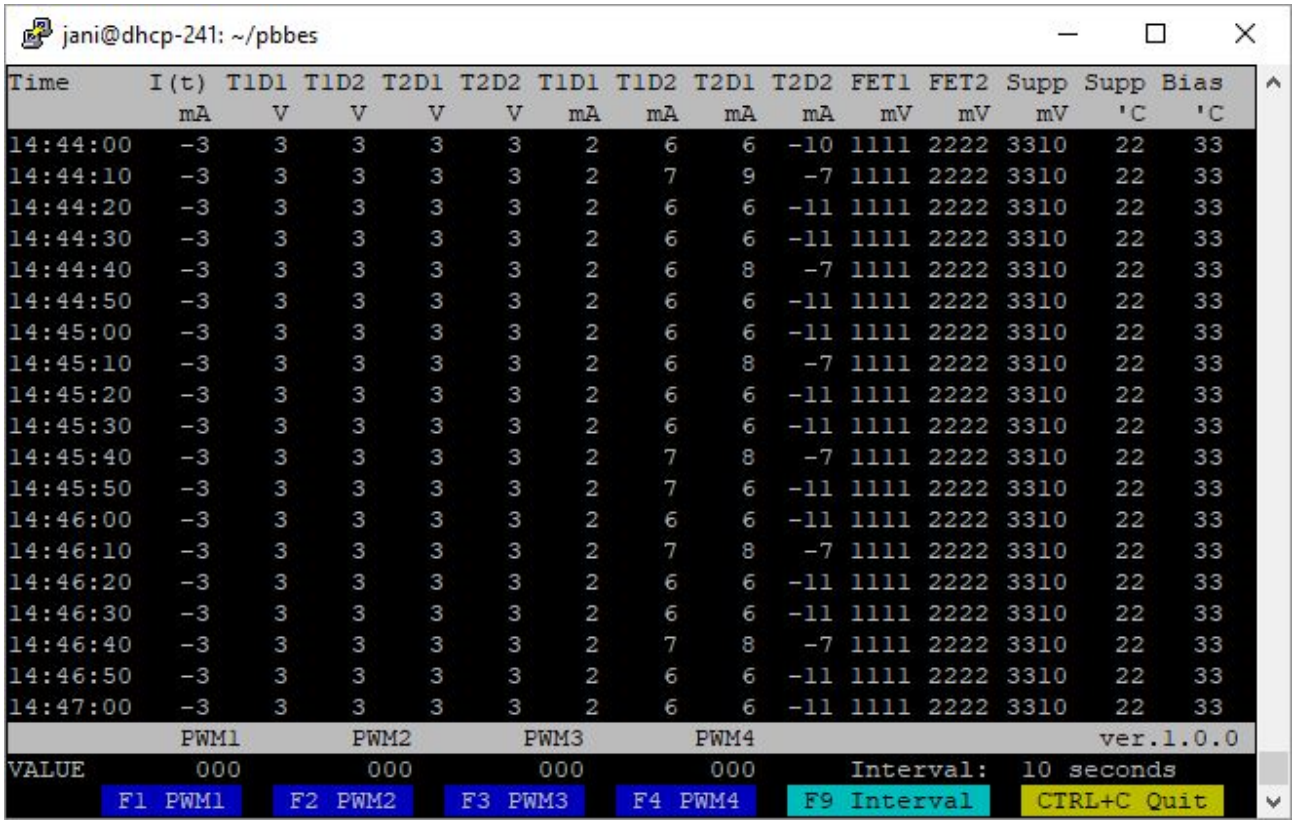
Picture 1 - Start-up screen for version 1.0.0

4.3 Main Screen

Main screen of the application performs two functions. Firstly, it displays the retrieved data and secondly, it allows the operator to issue PWM duty change commands.



Picture 2 - Concept model for Main Screen (NOTE: chapter numberig is out-dated)



Picture 3 - Main screen for version 1.0.0.

4.3.1 Measurement Data Display

This UI component will be a table-like entity that may or may not feature a scrollbar on the right, depending on the practical limitations of the Python library at the time of implementation.

- New rows will be entered at the bottom of the list and old rows will “raise up”.
- Header will be 2-row and static. First row will have short hand symbol of the column (4 characters maximum, unless not possible) and second row will have a unit descriptor (4 characters: V, mV, mA and ‘C). See below for an example.
- Time column will not show dates (CSV data will have date and time).
- Units will not be written into the table rows (they appear in the second row of the header).
- Time column will follow HH:MI:SS format (8 characters wide).
- All other value columns will contain data that is 4 digits wide and has no fractions. There will be no zero padding.

Time	I(t)	T1D1	T1D2	T2D1	T2D2	T1D1	T1D2	T2D1	T2D2	FET1	FET2	Supp	Supp	Bias
	mA	V	V	V	V	mA	mA	mA	mA	mV	mV	mV	'C	'C
22:02:43	24.4	553	683	127	460	54.5	43.0	86.7	55.5	125	41	123	698	340
22:02:53	01.3	482	611	217	258	81.8	85.1	22.7	22.2	540	694	496	821	225

Picture 4 - Table header matching the order of columns as specified in Chapter 3.5.

4.3.2 PWM Controls

A set of four PWM controls are provided, each 3 digits wide.

- Each PWM field will be populated with initial value retrieved from the bias board. The value is retrieved using command `PWMn?<LF>` (`n` = {1..4}), which returns an integer value `ddd<LF>OK<LF>` (`ddd` = {001...999}).
- Each input/display field is an integer of exactly 3 digits. Accepted range: [00]1 … 999. Software will automatically zero pad user input if necessary.
- Entering (ENTER key) a new value will initiate `PWMnSxxx<LF>` command. Replies are observed.
- Escape (ESC) will cancel value input.

4.3.3 Polling Interval

Simple control that allows setting the data polling interval in seconds.

- Field is an interger of 3 digits and displayed with space padding. Accepted value range: [00]1 … 999.
- Entering (ENTER key) will immediately enforce new polling interval.
- Escape (ESC) will cancel value input.

4.3.4 Stop Testing

Key combination CTRL-C will close the CSV file and terminate the program.

4.3.5 PWM State

This element has become obsolete in the current implementation. Original intent was to show the current/actual PWM duty value, while controls defined in 4.2.2 were merely the input fields. Current implementation merges these two.