Piet De Vaere

A Robust Sensing Platform
for High Alpine Sensor Networks

Semester Thesis SA-2017-15
March to June 2017

Tutor: Prof. Dr. Lothar Thiele
Supervisor: Dr. Jan Beutel
Supervisor: Tonio Gsell
Supervisor: Reto Da Forno
Abstract
This work documents the design of a wireless sensor network (WSN) platform for high alpine deployments. This new platform is intended to replace the current sensor nodes deployed by the PermaSense group, a research consortium devoted to developing and deploying WSNs in the high Alps. The new sensing platform is based on the Dual-Processor Platform (DPP) architecture. Meaning that application and communication tasks are mapped to different processors, interconnected via BOLT, an interprocessor communication mechanism that keeps power, clock and timing domains strictly decoupled. The new platform consists of two parts. Firstly, there is a communication module with a communication processor and BOLT chip. Secondly, there is a sensor interface board (SIB) with power management circuitry, sensor interface circuitry and an application processor. A prototype of the SIB has been developed in this thesis, and a communication module is currently in development. Initial tests show that the SIB is fully functional. Moreover, sleep current consumption shows an order of magnitude improvement over past PermaSense sensor platforms, and results for active current consumption are promising.
Contents

1 Introduction ........................................ 5

2 Background ........................................ 7
   2.1 PermaSense ..................................... 7
   2.2 Wireless Sensor Networks ...................... 7
   2.3 Sensor Interface Board ......................... 8
   2.4 Dual-Processor Platform ....................... 9
   2.5 Ultra-Low Power Electronics Design .......... 9

3 Design ............................................. 11
   3.1 General Considerations ......................... 11
   3.2 Requirements ................................... 11
   3.3 Architecture Selection ......................... 12
   3.4 Design Overview ................................ 13
      3.4.1 The Functional Diagram .................... 14
      3.4.2 The Power Diagram ......................... 16
   3.5 Board Input and Output ......................... 16
      3.5.1 Power Input ................................ 18
      3.5.2 Sensor Connector ........................... 18
      3.5.3 BOLT Connector ............................ 18
      3.5.4 Debug UART Header ......................... 18

4 Prototype ......................................... 21
   4.1 Testing ........................................ 21
      4.1.1 Test 1: Verification of Power Rails ...... 21
      4.1.2 Test 2: Functional Verification .......... 21
      4.1.3 Test 3: Power Trace ....................... 23
      4.1.4 Test 4: Deep Sleep Power Consumption .. 23
      4.1.5 Test 5: Sensor Rod Demo .................. 26
   4.2 Design Modifications ........................... 27

5 Conclusion and Future Work ....................... 31

Bibliography ....................................... 33

A Power Rail Dependencies ........................... 37

B Test UART Output .................................. 39

C Design Documents ................................ 41

D Schematic and PCB Overlay ....................... 43
Chapter 1

Introduction

The vision on wireless sensor networks (WSNs) has changed significantly over the past twenty years. Where Kahn et al. (1999) [1] presented a futuristic vision of cubic millimeter sized smart dust, modern work focuses on realistic, manufacturable designs. To this end, modern research is often supported by real world sensor deployments.

For example, the PermaSense [2] project studies practical WSN applications for harsh, high-mountain environments. In order to evaluate the real world performance of these WSNs, the PermaSense consortium operates a number of test networks in the Alps, one of which is shown in Figure 1.1. As these deployments have many common requirements, a standardized sensing platform was developed, and is used across different deployments.

This platform is build around a Tinynode processing and radio module, that plugs in to a so called sensor interface board (SIB). The former is an off-the-shelf component, while the later is a custom designed printed circuit board (PCB). The SIB’s function is to support the Tinynode, by providing it both power and an interface to a broad range of analog and digital sensors.

However, just as the WSN community has become wiser, so has the PermaSense team. The know-how gathered over the past years, together with the advance in technology has given rise to new sensor node requirements. Therefore, the time has come to develop a new sensing platform.

This new platform will be based on the Dual-Processor Platform (DPP) architecture [3]. The DPP architecture is a novel approach to WSN nodes, that imposes a strict separation between application (e.g. measurements) and communication (e.g. packet forwarding) tasks. It does so by mapping both types of tasks to different processors, and connecting them using BOLT [4]. BOLT is an interprocessor messaging mechanism that keeps the processor’ power, clock and timing domains strictly separated.

Similarly to the current sensing platform, the new one will consists of a SIB that mates with a radio module. The new SIB will have similar functionality to the old SIB, but will also have an on board processor to run application tasks. The radio module will only be used for communication tasks, much similar to a network interface card in commodity computing products.

This thesis documents the design of this new SIB. The design of the radio module is a parallel effort.
(a) Locations of the sensor nodes. Circles indicate sensor nodes, lines show the network links between them.

(b) A typical sensor node installation.

Figure 1.1: A PermaSense WSN deployment.
Chapter 2

Background

2.1 PermaSense

The PermaSense [2] consortium consists of a group of interdisciplinary researchers from multiple Swiss research institutions and companies. The self-stated goal of the consortium is to “develop, deploy and operate wireless sensing systems customized for long-term autonomous operation in high-mountain environments” [2].

Roughly speaking, PermaSense combines the interests of two groups: On the one hand, there are the electrical engineers. They are mainly interested in developing and deploying WSN infrastructure as a goal in itself. They ask questions like “How long does the sensor network stay operational?” or “How reliable is the data we gather?”. On the other hand, there are the natural scientists that interpret the gathered data. They ask questions like “What trends are visible in the data?” or “What is the influence of global warming on permafrost\(^1\)?”. High-alpine sensor networks combine the interests of both groups: the electrical engineers see the harsh environment as an excellent benchmark for the robustness of the sensor electronics, and the natural scientists get access to datasets that would be infeasible to acquire in any other way.

This thesis is written from the perspective of the engineers. Namely, it discusses the design and prototyping of a new sensor system.

2.2 Wireless Sensor Networks

Wireless sensor networks (WSNs) are — as the name suggests — wireless networks that consists of multiple sensor nodes. These sensor nodes are typically small and low power devices with one or more sensors attached to them. They are often deployed in hard to access locations, and should therefore function as independently as possible. Furthermore they are typically battery powered, requiring the developer to treat power as a scarce resource. For example, the sensor nodes deployed by the PermaSense team require a minimum of three years of operation on a single D sized battery.

Because of these extreme power requirements, WSN nodes can not use regular wireless technology like IEEE 802.11 (or, in more human terminology: “WiFi”). Instead they follow standards specifically designed for low power operation, most notably IEEE 802.15.4 (unfortunately, no human compatible name is available for this standard, but it forms the basis for the ZigBee [5] specification).

\(^1\) Permafrost is permanently frozen ground.
In order to span large distances using these low power radios, WSNs typically operate as a mesh network. Data packets are forwarded over this network towards a special node called the sink or base station. The sink logs this data to permanent storage, forwards the data over the internet (e.g. via a cellular connection), or does both. Because of the higher power requirements for the sink (mass storage devices or cellular radios are typically power hungry), they are usually installed in a location with grid power access, or outfitted with solar cells. A typical WSN topology is shown in Figure 2.1.

![Figure 2.1: Schematic diagram of a typical WSN setup.](image)

**2.3 Sensor Interface Board**

The current sensor networks deployed by the PermaSense team are based on the Shockfish Tinynode [6]. As the Tinynode is only a processing and communications module, it must be supported by additional hardware before it becomes a fully functional sensor node. Therefore, the PermaSense team has developed the sensor interface board.

The **sensor interface board** (SIB) is a custom build PCB containing all the circuitry needed to support the Tinynode. The most important components of the SIB are:

- Circuitry to interface the Tinynode with a wide range of analog and digital sensors.
- A D sized battery to power the sensor node.
- Power regulation circuitry to generate a number of (switchable) voltage rails.
- Sensors to monitor the node’s health: a temperature sensor, a humidity sensor and power monitors.
- User interface circuitry: LEDs, a reset button, a buzzer, ...

There are currently two generations of the SIB. Figure 2.2 shows how a sensor node based on the first generation SIB (also known as SIB1) looks like. The second generation SIB (SIB2) is an incremental update to the first one. The main difference is that it is designed for a larger metal housing.

This thesis documents the design of a third generation SIB (SIB3), designed to replace the tinynode with the **Dual-Processor Platform** (DPP).
2.4 Dual-Processor Platform

In the first two generations of the SIB, a Tinynode is used as a processing and communication module. Because a single microcontroller is used for both communication (e.g. forwarding packets from other nodes) and application tasks (e.g. performing measurements), resource interference between these two tasks is possible and even likely. As such interference can have a negative impact on performance, it is undesirable.

The Dual-Processor Platform (DPP) [3] is a wireless sensor node architecture that decouples the communication and application domain by running the tasks for each of them on a dedicated processor. These two processors — called the communication and application processor — are then connected to each other using BOLT [4]. BOLT is a processor interconnection mechanism that allows two processors to exchange messages in a fully asynchronous manner, while fully decoupling the time, power and clock domains. It does so by placing an active component, a message buffer between the communicating processors, and allowing them to read and write from this buffer according to their own schedule. We will refer to this active component as the BOLT chip.

Another advantage of the DPP architecture is that it allows for a modular design, where components can easily be exchanged. This is mainly thanks to the truly modular nature of a BOLT based interconnect. Figure 2.3 shows one of the early prototypes of the DPP architecture. The pluggable design of the BOLT module is a powerful demonstration of the modularity of the DPP design.

The third generation SIB will be based on the DPP architecture. The application processor will be embedded onto the SIB, and a BOLT compatible interface to connect to a communication module will be provide.

2.5 Ultra-Low Power Electronics Design

WSN deployments typically last for extensive periods of time. Because the nodes are usually battery powered, energy must be treated as an extremely scarce resource when designing a sensor node. Common techniques for ultra-low power electronics design are:

Using low power components When selecting components, care must be taken to select components with low active and quiescent current draws. For example, the popular LD1117V28 2.8 V linear voltage regulator [7] has a maximum quiescent current draw of
10 mA. At the same time, the TPS78228 regulator [8], designed for ultra-lower power operation, has a quiescent current draw of at most 1.4 μA. A difference of four orders of magnitude!

**Duty cycling** Most active electronic components have high current consumptions, even when they are just idle. For example, when the MSP432 microcontroller [9] is in running mode, it can easily consume a couple of mA, even when it is not doing any useful work. As most applications do not need all their resources all of the time, large energy savings can be accomplished by placing individual components in sleep mode. To come back to the example, when placed in its deepest sleep mode, the MSP432 consumes only 25 nA.

**Power gating** As not all electronic components have ultra-low power sleep modes, or to even further reduce power consumption, it can be useful to completely disable a power rail. Let us consider a power rail generated by the TPS78228 regulator discussed earlier. When the TPS78228 is placed in shutdown mode, no more power is supplied to its output pin. Thus, the power draw of the components it powers is completely eliminated. Furthermore, when placed in shutdown mode, the TPS78228 consumes merely 130 nA. Note however, that care must be taken that no power “leaks” into the disabled components, for example through an active signal line.

In the third generation SIB design, all of these techniques will be extensively used.
Chapter 3

Design

3.1 General Considerations

Past generations of PermaSense sensor nodes have all been based on a Tinynode that plugs in to a sensor interface board (SIB). Even though the new design will not be based on the Tinynode anymore, it will still consist of a SIB that connects to a communication module. This is discussed in more detail in Section 3.3. During this thesis, only the new SIB will be developed. The design of the new communication module is a parallel effort, that — at the time of this writing — has not been completed yet.

As the second generation sensor interface board (SIB) is a proven design, it was decided to use it as the basis for the design of the third generation SIB. However, the new SIB will be based on the Dual-Processor Platform (DPP) architecture, rather than to be build around a Tinynode. Based on good experiences in our research group, an MSP432 will be used as application processor.

Using this more modern processor allows for a significant design simplification. The reason for this is twofold. Firstly, the limited pin count of the Tinynode required many signals to share the same physical connections. Because of this, complex demultiplexing circuitry was required on the SIB. As the MSP432 has a plethora of pins, this will not be necessary any more. Secondly, because the MSP432 has a large amount of on board peripherals, many of the components on the second generation SIB (e.g. external universal asynchronous receiver/transmitters (UARTs)) can be eliminated.

The design of a new SIB is also an opportunity for design clarification. Through years of iterative design by multiple people, the documentation for the second generation SIB has lost some of its consistency and transparency. One of the goals of the new SIB design is to bring these back. Therefore, a strong focus is placed on creating clean schematics, clear code, consistent naming and a well documented design process throughout the project.

3.2 Requirements

At the beginning of this thesis project, a number of requirements for the new SIB were formulated. The most important ones are listed below. For a full overview, the reader is referred to the specification document referenced in Appendix C.

1. The SIB should be based on the Dual-Processor Platform architecture.
2. The SIB should be designed for the MSP432 processor.
3. The SIB should be capable of long-term storage.
4. It should be possible to reset the SIB when the sensor node box is sealed.
5. The SIB should provide four differential analog-to-digital converter (ADC) channels with variable gain.
6. The SIB should provide two single ended ADC channels with variable gain.
7. The SIB should be able to interface digital sensors over SDI-12 [10], RS-232 and optionally RS-485.
8. The SIB should have 6 digital outputs.
9. The SIB should have two switched 12 V outputs.
10. An input for an external 12 V DC supply should be available. This supply is most likely to be a solar powered 12 V battery system. The SIB should be able to internally generate a 12 V rail when no external 12 V supply is available.
11. The SIB should be powered by a D sized Li-SOCl$_2$ battery.
12. The SIB should be able to monitor its own health. That is, it should be able to monitor:
   (a) Its input voltages and battery current consumption.
   (b) Its internal temperature and humidity.
13. The new sensor platform should be usable as a drop-in replacement for current PermaSense deployments.

It should be noted that these are all soft requirements. If for some reason a requirement turns out to be unfeasible, or if deviating from it would significantly enhance the new platform, the requirements can be changed.

3.3 Architecture Selection

As stated in Requirement 1, the new SIB should be based on the DPP architecture. This means that the design must consist of an application processor (the MSP432 per Requirement 2), a communication processor (or radio) and a BOLT module. Because a three board design (as in Figure 2.3) is cumbersome from a mechanical perspective, it was decided to place two or more of these components on the same board.

Furthermore, in order to keep the design compatible with future radio modules, and to make it possible for the radio module to be used by multiple projects, it was decided not to place the radio circuitry on the SIB. Rather, a radio daughter board will plug in to the SIB, similar to how the Tinynode plugs in to the second generation SIB. This gives rise to the question, “Which components should be placed on the radio module, and which ones on the SIB?” Three possible answers, shown in Figure 3.1, were considered.

**Architecture A** (Figure 3.1(a)) places only the MSP432 on the SIB, and places both the BOLT and radio chip on the radio board. This has the advantage that the MSP432 can be closely integrated with the SIB, and that the interface between the two boards is simple. Furthermore, the radio board can be developed independently from the SIB.

**Architecture B** (Figure 3.1(b)) is very similar to architecture A, but places the BOLT chip on the SIB rather than on the radio board. Which one of these two architectures is better, is a rather philosophical question. Should BOLT be primarily seen as a feature of the radio, or of the application processor?
CHAPTER 3. DESIGN

(a) Only the MSP432 on the SIB.

(b) The MSP432 and BOLT on the SIB.

(c) Everything on the radio board.

Figure 3.1: Schematic representations of possible implementations of the DPP architecture.

Architecture C (Figure 3.1(c)) places all three components on the radio board. This architecture is the closest to the approach of the first and second generation SIB: no processing or communication is done on the SIB. Close integration with the MSP432 is still possible, but would require a complex interface between the two boards. Alternatively, multiple logical connections could be multiplexed over a simpler physical interface.

Considering the above, it was chosen to use Architecture A. There are three main reasons for this. Firstly, placing the MSP432 on the sensor interface board is preferable because it simplifies close integration of the microcontroller with the SIB. Secondly, Assuming that a standardized logical interface between the application processor (MSP432) and communication processor (radio) is defined, Architecture A allows the user to exchange the radio board (possible with one from another project), without having to reprogram the radio board or the SIB. Finally, it seems intuitively more natural to place the BOLT chip on the radio board. Facilitating asynchronous message passing is not one of the core features of the SIB. Rather, it should be seen as a feature provided by the radio board. Furthermore, as BOLT is still in active development, placing the BOLT chip on the radio board makes it easier to experiment with alternative versions.

3.4 Design Overview

The high level design for the new SIB is inspired by the design of the second generation board. However, there are some major differences between the old and new design. Probably the most significant of those is that the new SIB has an on board microcontroller, whereas the old design did not.

The two most important documents to understand the SIB3 design, are the functional diagram (Figure 3.2) and the power diagram (Figure 3.4).
3.4.1 The Functional Diagram

The functional diagram in Figure 3.2 shows that the MSP432 microcontroller is truly at the center of the design. The other major functional blocks in the design are:

The **Radio Board** is a daughter board of the SIB. As discussed in Section 3.3, it contains a BOLT chip and a radio module.

The **AD7708 ADC** is the main ADC used to digitize analog sensors. Being a Δ–Σ ADC, it has a lower sample rate than the MSP432 build-in successive approximation (SAR) ADC, but it offers a higher resolution (16-bit vs 14-bit) and has a programmable gain front end, partially fulfilling Requirements 5 and 6. The AD7708 has ten single ended inputs, that can be paired up to form differential inputs as needed. This fulfills Requirements 5 and 6 completely. A jumper allows the user to connect the analog ground for single ended measurements to digital ground, or to leave it floating. In the previous SIB design the same ADC was used. But because it was configured differently, and because it was also used to digitize signals internal to the sensor node, only six single ended channels were available to the user.

The **SDCARD slot** can be used to interface any microSD card in Serial Peripheral Interface (SPI) mode. This fulfills Requirement 3.

The **Sensirion SHT31** provides temperature and humidity measurements of the inside of the sensor node. This fulfills Requirement 12b.

The **Internal Serial Nr.** is a silicon serial number allowing a SIB to be uniquely identified. The 1-Wire based DS2401 is used, because previous SIBs and some of the PermaSense sensors use the same chip. Thus, in order to satisfy Requirement 13, the new SIB must also support 1-Wire.

The **Power Manager** is not a physical component, but a logical one. It represents the SIB’s ability to disable power rails as needed. It will be discussed in more detail in Section 3.4.2.

The **Power Monitors** allow the SIB to measure the battery voltage, battery current, and 12 V input voltage. They are digitized using the MSP432’s internal ADC, leaving all channels of the AD7708 available to the user. This fulfills Requirement 12a.

The **LEDs, Reset and Buzzer** circuitry provides information to the user, and allows the user to reset the SIB. Because of their high current draw, the LEDs should not be used during deployment. Therefore, a hardware jumper to break the current path through the LEDs is included in the design. In order to meet Requirement 4, there is not only a reset **button** on the board, but their is also a reed switch connected to the reset pin of the MSP432. This allows the user to reset the board by holding a magnet close to the reed switch. Even when the SIB is in a sealed enclosure. The buzzer can be used to signal something to the user (e.g. a reset), even when the sensor node box is sealed.

The **MAX3471, MAX3221 and 7400 Logic** are used to generate RS-485, RS-232 and SDI-12 [10] compatible voltage levels, respectively. As no off-the-shelf components exists to generate SDI-12 compatible voltage levels, 7400 logic has to be used. These serial lines are used to communicate with digital sensors, as required by Requirement 7.

The **UC_GPIO signals** consist of seven signal lines directly connected to the MSP432, providing general-purpose input/output (GPIO) to the user, and fulfilling Requirement 8. Furthermore, all of these GPIOs can be used as inputs to the MSP432’s internal 14-bit SAR ADC. Providing the user with up to seventeen ADC channels in total.

**ESD protection** is applied to all signal lines that leave the sensor node, and that will be used during deployment.
Figure 3.2: Functional diagram for the third generation SIB design. The naming in this diagram is followed throughout all design documents.
3.4.2 The Power Diagram

The power diagram in Figure 3.4 shows all voltage rails present on the SIB, and how they are generated.

Power enters the SIB at two points: via the battery, and via the 12VDC input. Because of Requirement 11, the battery is a Li-SOCl$_2$ cell. The Saft LSH 20 will be used, as this battery has performed excellently with the SIB2. It has a rated output voltage $V_{BAT}$ from 3.7 V to 3.2 V [11]. However, in practice, the voltage drops quickly once a value of about 3.4 V is reached, as can be seen in Figure 3.3. Both the battery voltage $V_{BAT}$ and battery current $I_{BAT}$ can be monitored by the MSP432. From the battery voltage, the following supply rails are generated:

- **2V8** is the always-on supply rail that feeds the MSP432.
- **2V8_SW** is the preferred voltage rail for components that the MSP432 should be able to power gate. However, not all components support supply voltages as low as 2.8 V, thus additional power rails are required.
- **VBAT_SW** is at the same voltage level as the $V_{BAT}$ rail, but can be disabled by the MSP432.
- **VSENSE** is generated by a variable low-dropout (LDO) regulator. It is an extra supply voltage for external sensors, and is needed because some of the older PermaSense sensors operate on an input voltage of 3.2 V. It can be disabled by the MSP432. Optionally the VSENSE rail can be joined with the 2V8_SW rail.
- **2V5,REF** is the reference voltage for analog measurements. It is connected to the AD7708’s external reference input, and can be disabled by the MSP432.
- **12V,GEN** is the internally generated 12 V rail, as required by Requirement 10. It can be used to power SDI-12 equipment when no external 12 V rail is available. It can be enabled by the MSP432.

The user should select whether he wants to use the external 12 V input ($12V_IN$) or the internally generated 12 V rail ($12V_GEN$) before deployment by setting a jumper. This prevents power hungry equipment from accidentally draining the internal battery.

From the selected 12 V rail — referenced as $12V_OUT$ —, the $5V0$ rail is generated. This rail is used to drive the signal line of the SDI-12 bus [10]. It should be noted that generating the $5V0$ rail from the internally generated $12V_GEN$ rail is power inefficient. However, the $5V0$ rail is only used to drive the SDI-12 data line. According to the SDI-12 specification [10], this data line has a ground resistance of at least 80 kΩ$^1$. This results in a maximal current draw of:

$$I = \frac{5V}{80k\Omega} < 64\mu A$$

Which is small enough that even low conversion efficiencies are tolerable.

The $12V_OUT$ rail is also used to generate two switched 12 V outputs: $12V_SW1$ and $12V_SW2$, as required by Requirement 9.

Appendix A provides an overview of what power rails should be active for which subsystems to be functional.

3.5 Board Input and Output

The SIB3 has a number of input and outputs, the most important of which will be discussed in this section.

$^1$Assuming two devices are present on the bus.
Figure 3.3: Plot showing how the battery voltage of a sensor node deployed on the Matterhorn evolves over time. It can be seen that once the battery voltage reaches 3.4 V, it drops off rapidly. Around the 57-day mark the battery is replaced. Other sensor nodes show the same behaviour. The zero point on the time axis does not correspond to the start of the deployment.

Figure 3.4: Power diagram for the third generation SIB design. The naming in this diagram is followed throughout all design documents.
3.5.1 Power Input

The SIB3 has four possible power inputs. Firstly, there is the battery, which is intended as the main power source during deployment. The board has a D sized battery holder, and guarantees support for input voltages in the $3.4\,\text{V}$ to $3.7\,\text{V}$ range. The battery is protected by a $500\,\text{mA}$ self resetting fuse, and has no reverse polarity protection.

Secondly, there is the $\text{VBAT}$ header next to the battery holder. This header is connected to the same power rail as the battery, and is intended to power the board while being serviced. There is no input protection present on this header, but a placeholder footprint for a fuse or series diode is provided.

Thirdly, there is the $\text{VBAT}$ pin on the 40-pin sensor connector (see Section 3.5.2). While this pin is primarily intended as a power output, it can also be used as an input. When the SIB is powered through this pin, the power switch is not functional. This pin has electrostatic discharge (ESD) protection, but no fuse or reverse polarity protection.

Finally, there is the $12\,\text{V}_{\text{IN}}$ pin on the 40-pin sensor connector (see Section 3.5.2). It can be used as a $12\,\text{V}$ input (alternatively, the SIB can generate a $12\,\text{V}$ rail internally). Support for voltages in the $9.6\,\text{V}$ to $16\,\text{V}$ range is guaranteed. This input is protected by a slow acting $2\,\text{A}$ fuse, and has ESD protection.

3.5.2 Sensor Connector

The largest connector on the SIB3 is the 40-pin sensor connector. Its pinout is shown in Table 3.1. All power rails, the serial buses intended to communicate with external sensors, the ADC channels and the user GPIO ($\text{UC_GPIO}$) are all available on this connector. All of the pins are ESD protected.

Because of the series resistors in the ESD protection circuit, the current through the $12\,\text{V}_{\text{IN}}$, $12\,\text{V}_{\text{OUT}}$, $12\,\text{V}_{\text{SW1}}$ and $12\,\text{V}_{\text{SW2}}$ pins should not exceed $0.5\,\text{A}$ continuously or $2\,\text{A}$ at a $10\%$ duty cycle. The current through any of the other power rail pins should not exceed $100\,\text{mA}$. All these values are expressed per pin.

Pins from this connector are intended to be wired to a ruggedized and waterproof connector on the outside of the sensor node housing (i.e. the “Sensor connector” in Figure 2.2).

3.5.3 BOLT Connector

The BOLT connector is intended to connect the SIB to the radio module. However, as the design of this module is not yet know, and because a physical standard for the BOLT connector is still a work in progress, the first prototype of the SIB3 is designed to connect to the BOLT prototype shown in Figure 2.3. For a next tapeout of the design, the connector should be replaced. The current design provides ample of board space to place this new connector.

The pinout for the current BOLT connector is shown in Table 3.2. The connector provides the always on $2\,\text{V}_{\text{8}}$ rail, as BOLT modules are supposed to handle their own power management.

3.5.4 Debug UART Header

As its name suggests, the debug UART header can be used to connect to the debug UART. The UART header has a ground, $\text{Tx}$, $\text{Rx}$ and reset pin. This last pin can be used to reset the microcontroller when a UART bootloader is used.
Table 3.1: Pinout of the 40-pin sensor connector

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SDI12_DATA</td>
<td>2</td>
<td>12V_IN</td>
</tr>
<tr>
<td>3</td>
<td>RS485_A</td>
<td>4</td>
<td>12V_OUT</td>
</tr>
<tr>
<td>5</td>
<td>RS485_B</td>
<td>6</td>
<td>12V_SW1</td>
</tr>
<tr>
<td>7</td>
<td>RS232_RX</td>
<td>8</td>
<td>12V_SW2</td>
</tr>
<tr>
<td>9</td>
<td>RS232_TX</td>
<td>10</td>
<td>5V0</td>
</tr>
<tr>
<td>11</td>
<td>AD7708_CH7</td>
<td>12</td>
<td>GND</td>
</tr>
<tr>
<td>13</td>
<td>AD7708_CH8</td>
<td>14</td>
<td>2V5_REF</td>
</tr>
<tr>
<td>15</td>
<td>AD7708_CH1</td>
<td>16</td>
<td>GND</td>
</tr>
<tr>
<td>17</td>
<td>AD7708_CH2</td>
<td>18</td>
<td>VSENSE</td>
</tr>
<tr>
<td>19</td>
<td>AD7708_CH3</td>
<td>20</td>
<td>VBAT</td>
</tr>
<tr>
<td>21</td>
<td>AD7708_CH4</td>
<td>22</td>
<td>2V8_SW</td>
</tr>
<tr>
<td>23</td>
<td>AD7708_COM</td>
<td>24</td>
<td>2V8</td>
</tr>
<tr>
<td>25</td>
<td>AD7708_CH5</td>
<td>26</td>
<td>VBAT_SW</td>
</tr>
<tr>
<td>27</td>
<td>AD7708_CH10</td>
<td>28</td>
<td>GND</td>
</tr>
<tr>
<td>29</td>
<td>AD7708_CH9</td>
<td>30</td>
<td>GND</td>
</tr>
<tr>
<td>31</td>
<td>AD7708_CH6</td>
<td>32</td>
<td>GND</td>
</tr>
<tr>
<td>33</td>
<td>1WIRE1</td>
<td>34</td>
<td>UC_GPIO7</td>
</tr>
<tr>
<td>35</td>
<td>UC_GPIO6</td>
<td>36</td>
<td>UC_GPIO5</td>
</tr>
<tr>
<td>37</td>
<td>UC_GPIO4</td>
<td>38</td>
<td>UC_GPIO3</td>
</tr>
<tr>
<td>39</td>
<td>UC_GPIO2</td>
<td>40</td>
<td>UC_GPIO1</td>
</tr>
</tbody>
</table>

Table 3.2: Pinout of the BOLT connector

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2V8</td>
<td>2</td>
<td>GND</td>
</tr>
<tr>
<td>3</td>
<td>BOLT_FUTURE_USE</td>
<td>4</td>
<td>BOLT_C_IND</td>
</tr>
<tr>
<td>5</td>
<td>BOLT_ACK</td>
<td>6</td>
<td>BOLT_TIM_REQ</td>
</tr>
<tr>
<td>7</td>
<td>BOLT_REQ</td>
<td>8</td>
<td>SPI_BOLT_SCLK</td>
</tr>
<tr>
<td>9</td>
<td>BOLT_MODE</td>
<td>10</td>
<td>SPI_BOLT_MISO</td>
</tr>
<tr>
<td>11</td>
<td>BOLT_IND</td>
<td>12</td>
<td>SPI_BOLT_MOSI</td>
</tr>
</tbody>
</table>
Chapter 4

Prototype

In order to verify the new SIB design, parts for five boards were ordered. These boards are a four layer design, and the schematic and PCB overlay can be found in Appendix D. One of these boards was assembled, and used to run a number of tests (see Section 4.1). The assembled prototype, can be seen in Figure 4.1. The results of these tests gave rise to a number of design changes that should be incorporated in a next iteration of the SIB3. These are listed in Section 4.2.

4.1 Testing

4.1.1 Test 1: Verification of Power Rails

Being faithful to the golden rule of electronics debugging ("they shall measure voltages"), the first test performed on the new SIB was a check of all voltage rails. For this test, all power rails were enabled by software, and the voltages on them measured.

The results are shown in Table 4.1. All measured voltages are well within their specification. However, two measurements are noteworthy: the values of the 2V5_REF and the 12V_GEN rail.

The 2V5_REF rail has a deviation of $-0.003$ V from the design value of $2.500$ V. On first sight, this appears as a large deviation for a voltage reference, but consulting the datasheet of the REF192G [12], shows that the initial accuracy is $2.490$ V to $2.510$ V. Thus, the measured value is well within specification. If required, the REF192G can be replaced by the pin compatible REF192E, which has an initial accuracy of $2.498$ V to $2.502$ V.

The 12V_GEN rail measures at $11.53$ V, which is on the low side for an unloaded $12$ V rail. However, the boost converter that generates the 12V_GEN rail, is set to output $11.50$ V by design. This value was copied from the SIB2 design. For a next iteration of the SIB3 design, this should be changed to $12$ V.

4.1.2 Test 2: Functional Verification

In order to functionally verify all parts of the design, a test application that interacts with all functional blocks was written. This code was written with the sole purpose of verifying the hardware design. It was not written to properly handle errors or to be power efficient, and should therefore never be used in anything that is even remotely close to production. However, the code can be used as guidance when writing a production grade application.
Figure 4.1: The first prototype of the third generation SIB.
The pseudocode for the test application is shown in Listing 4.1. All UARTs run at 115 200 baud, except for SDI-12, which runs at 1200 baud. The debug LEDs were used to display a counter keeping track of the program state. This counter was also written to the user GPIO pins (UC_GPIO). After some small hardware modifications (see Section 4.2), all subsystems where fully functional. The output from the debug UART is provided in Appendix B.

4.1.3 Test 3: Power Trace

In order to get a first impression of the new SIB’s power consumption, a power trace of the test application from Section 4.1.2 was recorded. For this measurement the debug LEDs were disabled with a jumper. The test setup is shown in Figure 4.2. The SIB is connected to a bench power supply with a RocketLogger\(^1\) performing a series current measurement. Furthermore, the RocketLogger’s digital inputs are connected to the UC_GPIO pins. This makes it possible to link the current measurements to the active program state.

The trace in Figure 4.3 shows that during operation, current consumption is relatively stable around 10 mA (the red dotted line). There are however, a number of clear increases in current draw:

- **When power is switched on** there is a large current surge up to about 500 mA. This is because many capacitors on the board need to be charged. Because the system is not yet running when this peak occurs, it is not a problem for proper operation of the SIB. When the SIB’s power source can not supply such a high current, the capacitors will charge slower, resulting in a wider but lower peak.

- **When the power rails are enabled** there is another current peak, of about 160 mA. This peak is also caused by charging capacitors. The height of the peak can be reduced by switching on power rails in steps, instead of enabling them all at once.

- **When the buzzer is sound** current rapidly varies from 20 mA to 35 mA.

- **When the SD card is accessed** current peaks up to 60 mA are measured.

- **When the RS-232 transmitter is enabled** current rises to about 35 mA.

This shows that the most power hungry components are the buzzer, SD card and RS-232 transceiver. Thus, when developing an application for the SIB, extra care must be taken to minimize their active times.

The trace also shows that when all switchable power rails are disabled, the current draw drops to approximately 5 mA (the green dotted line). This is the current drawn by the microcontroller when executing idle instructions. As the microcontroller does not enter a low power state, this does not correspond to the standby current of the board. What this measurement does show us, is that enabling all power rails adds approximately 5 mA to the boards power consumption. This can be seen as a worst case increase, as in production grade applications, only the required voltage rails will be enabled.

4.1.4 Test 4: Deep Sleep Power Consumption

The test in Section 4.1.3 provides insight in the power consumption of the various board components when active. However, as the SIB will be heavily duty cycled during deployments, it is essential that the board has a low current consumption during (deep) sleep.

\(^1\)The RocketLogger is an in house measurement device that can be used for precision, long term, voltage and current measurements. It also has digital inputs that can be sampled alongside the analog channels.
Table 4.1: Measured and design values of the power supply rails.

<table>
<thead>
<tr>
<th>Rail</th>
<th>Design value</th>
<th>Measured value</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBAT</td>
<td>3.4 V to 3.7 V</td>
<td>3.595 V</td>
<td>+0.000 V</td>
</tr>
<tr>
<td>VBAT_SW</td>
<td>3.4 V to 3.7 V</td>
<td>3.591 V</td>
<td>+0.000 V</td>
</tr>
<tr>
<td>2V5_REF</td>
<td>2.500 V</td>
<td>2.497 V</td>
<td>−0.003 V</td>
</tr>
<tr>
<td>VSENSE</td>
<td>3.161 V</td>
<td>3.176 V</td>
<td>+0.015 V</td>
</tr>
<tr>
<td>2V8_SW</td>
<td>2.800 V</td>
<td>2.822 V</td>
<td>+0.022 V</td>
</tr>
<tr>
<td>2V8</td>
<td>2.800 V</td>
<td>2.812 V</td>
<td>+0.012 V</td>
</tr>
<tr>
<td>5V0</td>
<td>5.000 V</td>
<td>5.000 V</td>
<td>+0.000 V</td>
</tr>
<tr>
<td>12V_GEN</td>
<td>11.50 V</td>
<td>11.53 V</td>
<td>+0.03 V</td>
</tr>
<tr>
<td>12V_SW1</td>
<td>11.50 V</td>
<td>11.53 V</td>
<td>+0.03 V</td>
</tr>
<tr>
<td>12V_SW2</td>
<td>11.50 V</td>
<td>11.53 V</td>
<td>+0.03 V</td>
</tr>
</tbody>
</table>

Listing 4.1: Pseudocode for the functional verification program.

```python
1 Main:
2   Initialize all GPIO
3   Initialize UART
4   Initialize BOLT
5   Sound buzzer for 1000 ms
6   While True:
7       Disable all power rails
8       Wait 100 ms
9       Read power monitors
10      Enable all power rails
11      Read power monitors
12      Initialize SD card
13      Read out SHT31 (on board temperature and humidity)
14      Calibrate Channel 1 from the AD7708 (external ADC)
15      Read out Channel 1 from the AD7708 (external ADC)
16      Read out internal serial number over 1-Wire
17      Send test string over RS-232
18      Send test string over RS-485
19      Send test string over SDI-12
20      Send test string over BOLT
21      Wait 500 ms
22
23 RS-232, RS-485 and SDI-12 interrupt routine:
24   Read character from bus
25   If character == '\n':
26     Print out characters over debug UART
```
CHAPTER 4. PROTOTYPE

Figure 4.2: Test setup used to generate the power trace in Figure 4.3. A RocketLogger measures
the current consumption of the SIB, and records the program state through the UC_GPIO pins.

Figure 4.3: Power trace of the SIB3 functional test application in Listing 4.1. Three phases of
the application are shown: initialization (purple), first iteration of loop (white) and the start of the
second loop iteration (red). Power was supplied to the board from a bench top power supply, that
was switched on around the 200 ms mark, as indicated in the figure.
It is therefore useful to quantify the minimal power consumption of the new SIB. That is, the current consumed by the board when it is placed in its lowest power mode. To that end, a test program that disables all power rails and places the MSP432 in its lowest power mode (LPM4.5) was uploaded to the board. Next all debugging equipment was disconnected, and the board’s power consumption measured using a precision current meter. In order to better simulate deployment conditions, a sensor rod with four thermistors was attached. This setup is shown in Figure 4.4.

Once the microcontroller entered in to LPM4.5 mode, the board’s power consumption dropped below 2 µA. Although this is a very promising result, it should be mentioned that the LPM4.5 mode will probably not be used during deployment, as the MSP432 can only be woken up from this mode by an external signal. We can however, use this measurement to predict the board’s current consumption in other low power modes.

Two low power modes are likely to be used extensively in deployment applications: LPM3.5 and LPM0.

LPM3.5 is the lowest power mode still allowing wake ups from the MSP432’s internal real time clock (RTC). Therefore, this mode is ideal to be used while waiting between measurement cycles. According to the MSP432’s datasheet [9], the rated current consumption for this power mode is 630 nA.

LPM0 is the highest power sleep mode. It allows the microcontroller to be waken up by most of its internal peripherals. The rated current consumption for this power mode is 500 µA. This assumes a clock speed of 8 MHz and that all peripherals are inactive. Enabling a SPI module adds 40 µA to this value. Lowering the clock speed reduces these values further.

The MSP432’s datasheet [9] states that the microcontroller’s current consumption in LPM4.5 is 25 nA. Given that placing the MSP432 in LPM3.5 instead consumes about 600 nA more, and given the measurement value above, this suggests that the SIB3’s current draw between measurement cycles will be well below 3 µA. For comparison, the SIB1 has a power consumption of 26 µA while sleeping [14]. No data for SIB2 is available.

During the functional test application from Section 4.1.2, almost all of the processors’ time is spent performing idle computation (i.e. time delay loops, flag polling or pin polling). In a production grade application this can be eliminated by configuring an interrupt and by placing the MSP432 in LPM0 until this interrupt fires. This means that the average current consumption of the MSP432 will effectively drop from the 5 mA measured in Section 4.1.3, to approximately 540 µA. It is also known from Section 4.1.2 that the current consumption of all peripherals (except the buzzer, SD card and RS-232 transceiver) is upper bounded by 5 mA. Thus, the expected power consumption of the SIB3 during data acquisition can be upper bounded to $5 \text{ mA} + 540 \mu \text{A} < 6 \text{ mA}$.

It is important to remember that this is only an upper bound, and not the actual expected current consumption. The SIB1 consumes about 2 mA when performing its acquisition cycle (excluding RS-232 and ADC operation) [14].

### 4.1.5 Test 5: Sensor Rod Demo

This final test was designed to be as close as possible to a deployment situation. It is based on the functional test in Section 4.1.2, but with a number of modifications:

- The SIB is now battery powered, as this will be the power source used during deployment.
- A sensor rod is attached to the SIB. The sensor consists of a plastic rod with four metal rings, each thermally coupled to a thermistor. This type of sensor is used by the PermaSense team to measure temperatures at different depths beneath the surface. The schematic and wiring of the sensor are shown in Figure 4.5(b).
• Instead of digitizing one channel from the AD7708, all four channels connected to the sensor rod are digitized.

• The following information is periodically printed over BOLT:
  - The humidity and temperature measured with the SHT31 sensor on the SIB.
  - The SIB’s battery voltage and current.
  - The four temperatures measured by the sensor rod.

The full setup is shown in Figure 4.5. To verify proper functionality, the data transmitted over the debug UART and BOLT serial lines is monitored by a desktop computer. The BOLT output of this program is shown in Listing 4.2, the UART output in Appendix B. After the first iteration of the program loop, the T1 metal ring was cooled down with freezer spray. It can be seen in the BOLT output (in Listing 4.2) that the measured temperature only decreases gradually. This is due to the large thermal mass of the metal rings on the sensor.

Listing 4.2: BOLT output of the sensor rod demo. Edited for readability.

```
1 Node Health (A): VBAT = 3.63 V IBAT = 13.05 mA
2 Node Health (B): T = 24.51 deg. C H = 45.89 RH
4
5 Node Health (A): VBAT = 3.63 V IBAT = 15.36 mA
6 Node Health (B): T = 24.41 deg. C H = 45.88 RH
8
9 Node Health (A): VBAT = 3.62 V IBAT = 9.67 mA
10 Node Health (B): T = 24.09 deg. C H = 45.62 RH
12
13 Node Health (A): VBAT = 3.62 V IBAT = 9.58 mA
14 Node Health (B): T = 24.02 deg. C H = 45.98 RH
15 Sensor: T1: 6.04 T2: 22.85 T3: 24.03 T4: 24.15
16
17 Node Health (A): VBAT = 3.62 V IBAT = 13.93 mA
18 Node Health (B): T = 24.14 deg. C H = 46.92 RH
19 Sensor: T1: 0.70 CT2: 22.40 T3: 23.93 T4: 24.13
```

4.2 Design Modifications

Assembling and testing the first SIB3 prototype uncovered a number of changes that should be made to the SIB3 design. There are only two critical changes, which are listed below. A full list can be found in the design documents referenced in Appendix C.

• The vias under the SD card slot should be removed. They can cause short circuits with the contacts that mate up with the SD card. The SD card slot should be rotated 180 degrees.

• The buzzer should be replaced with the buzzer from the SIB2 design. It should be driven by a low side MOSFET.
Figure 4.4: Test setup used to measure the deep sleep power consumption of the new SIB. The SIB is powered from a bench power supply, and its supply current is measured by a precision current meter. A thermistor sensor rod is attached to the 40-pin sensor header.
(a) A thermistor sensor rod is attached to a battery powered SIB. Debug UART and BOLT output is monitored with a desktop computer.

(b) Schematic and wiring of the thermistor sensor rod.

Figure 4.5: Test setup for the sensor rod demo.
Chapter 5

Conclusion and Future Work

The goal of this thesis was to design a robust sensing platform for high alpine wireless sensor networks. This design will serve to replace the current — outdated — platform used by the PermaSense consortium. However, as the current PermaSense platform is a proven design, it was used as a reference for the new one.

It was chosen to follow a similar approach as the current platform: a communication module plugs in to a sensor interface board (SIB). The later provides power and a sensor interface to the former. However, where in previous designs the communication module was also used for application tasks (e.g. reading out sensors), the new design follows the Dual-Processor Platform (DPP) architecture. The DPP architecture decouples communication and application tasks by mapping them to different processors, that communicate using BOLT. BOLT is a processor interconnect allowing asynchronous message passing while keeping the power, clock and time domains of the communicating processors strictly decoupled. More concretely, the new SIB includes a MSP432 microcontroller that handles all application tasks. The communication module hosts the communication processor and the BOLT chip.

This design does not only limit resource interference between the communication and application domains, but it also allows closer integration between the application processor and the application itself. That is, by embedding the MSP432 on to the SIB, it can be closely coupled in to the power management and sensor interface circuitry, resulting into an overall design simplification. Moreover, by keeping all application depended logic on the SIB, a single radio module design can be used across different projects.

A prototype SIB using this new architecture was designed, manufactured and tested. Initial tests show that the design is fully functional and shows excellent power performance. The new design improves sleep current consumption by an order of magnitude, and first measurements suggest that the active current consumption will at least match that of the old design.

The design of a communication module is a parallel effort that is still ongoing. Once it is completed, it can be combined with the SIB. Nevertheless, because the new sensing platforms follows the DPP architecture, the communication and application domains are fully decoupled. This means that code development for the new SIB does not have to wait for the completion of the communication module.
Bibliography


List of Acronyms

Acronyms

ADC analog-to-digital converter
DPP Dual-Processor Platform
ESD electrostatic discharge
GPIO general-purpose input/output
LDO low-dropout
LED light emitting diode
MOSFET metal-oxide-semiconductor field-effect transistor
PCB printed circuit board
RTC real time clock
SAR successive approximation
SIB sensor interface board
SPI Serial Peripheral Interface
UART universal asynchronous receiver/transmitter
WSN wireless sensor network
Appendix A

Power Rail Dependencies

Table A.1 provides an overview of which power supply rails should be active when using a SIB3 subsystem.

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Supply Rail</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOLT</td>
<td>None</td>
</tr>
<tr>
<td>AD7708</td>
<td>2V8_SW</td>
</tr>
<tr>
<td>SDCARD</td>
<td>2V8_SW</td>
</tr>
<tr>
<td>SHT31</td>
<td>2V8_SW</td>
</tr>
<tr>
<td>Internal Serial Nr.</td>
<td>2V8_SW</td>
</tr>
<tr>
<td></td>
<td>VBAT_SW</td>
</tr>
<tr>
<td>LEDs</td>
<td>None</td>
</tr>
<tr>
<td>Buzzer</td>
<td>None</td>
</tr>
<tr>
<td>UART_DEBUG</td>
<td>None</td>
</tr>
<tr>
<td>RS485</td>
<td>2V8_SW</td>
</tr>
<tr>
<td></td>
<td>VBAT_SW</td>
</tr>
<tr>
<td>RS232</td>
<td>2V8_SW</td>
</tr>
<tr>
<td></td>
<td>VBAT_SW</td>
</tr>
<tr>
<td>SDI12</td>
<td>2V8_SW</td>
</tr>
<tr>
<td></td>
<td>12V_GEN$^1$</td>
</tr>
<tr>
<td></td>
<td>5V0</td>
</tr>
</tbody>
</table>

$^1$Unless 12 V is supplied through the 12V_IN rail.
$^2$The right 12 V supply rail should be selected with a jumper.
Appendix B

Test UART Output

During the functional system test (Section 4.1.2) and sensor rod demo (Section 4.1.5) the messages on the debug UART were recorded. Below the output of a single loop cycle of the sensor rod demo program is given. The output of the functional system test is largely similar.

At the top of the test loop
1110 main.c 64: POWER: Disabling all power rails
2115 driver.c 864: VBAT: 3.639488 V
2119 driver.c 865: IBAT: 8.941650 mA
2123 driver.c 866: 12V_IN: 0.000000 V
2126 main.c 74: POWER: Enabling all power rails
2231 driver.c 864: VBAT: 3.629570 V
2235 driver.c 865: IBAT: 15.797933 mA
2239 driver.c 866: 12V_IN: 0.000000 V
2243 sib3.c 151: SDCARD: Waiting for SD card
2247 sib3.c 153: SDCARD: Card inserted
2353 mmc.c 225: SDCARD: card initialized
2357 sib3.c 171: SDCARD: Card initialization SUCCESS; status: 0
2376 sib3.c 705: SHT31 temperature: 24.511330 deg. C
2381 sib3.c 706: SHT31 humidity: 45.876249 perc. RH
3115 sib3.c 478: AD7708: CH1 zero scale call done
3494 sib3.c 493: AD7708: CH1 full scale call done
3600 sib3.c 508: AD7708: CH1 measurement done
3605 sib3.c 556: AD7708: CH1 has value: 0.692901 V
3610 sib3.c 134: THERMISTOR: temperature = 24.139780
3989 sib3.c 478: AD7708: CH2 zero scale call done
4368 sib3.c 493: AD7708: CH2 full scale call done
4474 sib3.c 508: AD7708: CH2 measurement done
4479 sib3.c 556: AD7708: CH2 has value: 0.691221 V
4484 sib3.c 134: THERMISTOR: temperature = 24.224386
4863 sib3.c 478: AD7708: CH3 zero scale call done
5242 sib3.c 493: AD7708: CH3 full scale call done
5348 sib3.c 508: AD7708: CH3 measurement done
5353 sib3.c 556: AD7708: CH3 has value: 0.692003 V
5358 sib3.c 134: THERMISTOR: temperature = 24.185053
5738 sib3.c 478: AD7708: CH4 zero scale call done
6116 sib3.c 493: AD7708: CH4 full scale call done
6222 sib3.c 508: AD7708: CH4 measurement done
6227 sib3.c 556: AD7708: CH4 has value: 0.692120 V
6233 sib3.c 134: THERMISTOR: temperature = 24.179152
6244 main.c 107: 1WIRE ROM: ff 02 84 19 00 00 (fcode 01)
6250 sib3.c 600: RS232: Sending test string
<<< note: no data was received over RS-232 because their was no loopback on the Rx and Tx lines. >>>

6357 sib3.c 629: RS485: Sending test string

6364 main.c 417: RSXXX data received: >>RS485 RS485 RS485 RS485

6471 sib3.c 573: SDI12: Sending test string

6504 main.c 388: SD12 data received: >>SDI12 SDI12 SDI12 SDI12

6610 bolt.c 289: BOLT: send SUCESSFULL

6614 bolt.c 289: BOLT: send SUCESSFULL

6619 bolt.c 289: BOLT: send SUCESSFULL

6623 bolt.c 289: BOLT: send SUCESSFULL
Appendix C

Design Documents

This appendix provides an overview of all relevant design documents for the SIB3, and where to find them.

Design data can be found in two internal SVN repositories (only accessible for members of TIK).

The code for the MSP432 can be found in the https://svn.ee.ethz.ch/DualProcessorPlatform/code/application_processor/MSP432/sib3_test/ repository. All other design data can be found in the https://svn.ee.ethz.ch/tecstuds/devaerep/ repository.

Important files and directories in the /tecstuds/devaerep/ repository are:

- **altium/** contains the latest altium files.
- **board_shape/SIB_PCB_Board_2.dxf** contains the board outline.
- **design_process_notes/SIB3_design_process_notes.tex** contains notes about the design process of the SIB3. Some, but not all of these notes in this document are incorporated in this thesis.
- **DS/** contains datasheets of the various components in the design. It also contains relevant application notes.
- **BOM/** contains the bill of materials and order lists.
- **BOM/sib3_placement_BOM.xlsx** provides a mapping between the reference designators and a CMP NR. All components in storage have their CMP NR marked on them, and are stored sorted on CMP NR.
- **papers/** contains some relevant papers.
- **rev1_design/** contains a snapshot of the design files for revision 1.
  - **rev1_design/sib3_rev1.pdf** is the schematic.
  - **rev1_design/sib3_rev1_bottom_overlay.pdf** is the bottom layer printed circuit board (PCB) overlay.
  - **rev1_design/sib3_rev1_top_overlay.pdf** is the top layer PCB overlay.
- **required_changes.txt** lists everything that should be changed for a next board version.
- **rocketlogger/** contains power trace data recorded with the RocketLogger. Data files are named with a timestamp of the measurement, followed by the SVN commit number of the
code was active on the board. These commit numbers refer to the DualProcessorPlatform repository.

rocketlogger/plot_power_trace.m is a handy MATLAB script to plot the power traces recorded with the RocketLogger.

serial_out/ contains debug UART and BOLT outputs of the tests run on the board.

specification/functional_diagram.dia is the functional diagram of the SIB3.

specification/power_concept.dia is the power diagram of the SIB3.

specification/URS_SIB3.doc is the specification document on which the SIB3 design is based.

uc_pin_assignment/ contains the pin mapping of the MSP432. .pinmux files should be opened with the TI pinmux tool.
Appendix D

Schematic and PCB Overlay
12V switching

12V external

12V IN

12V SWITCHING

12V internal

12V IN

12V SWITCHING

External battery supply

External battery holder

Internal battery holder

Battery connectors

Terminal for external power supply

The FDV304P has a maximum V_GS of 14.4V. Hence a fix resistor divider and the zener: Under normal 12V operation no current should flow through the zener.

Voltage and Current monitors

The FDV304P has a maximum V_GS of 14.4V. Hence a fix resistor divider and the zener: Under normal 12V operation no current should flow through the zener.

PTC

PTC Resettable Fuses

500mA

Battery connectors

Terminal for external power supply

The FDV304P has a maximum V_GS of 14.4V. Hence a fix resistor divider and the zener: Under normal 12V operation no current should flow through the zener.

Voltage and Current monitors

The FDV304P has a maximum V_GS of 14.4V. Hence a fix resistor divider and the zener: Under normal 12V operation no current should flow through the zener.

PTC

PTC Resettable Fuses

500mA

Battery connectors

Terminal for external power supply

The FDV304P has a maximum V_GS of 14.4V. Hence a fix resistor divider and the zener: Under normal 12V operation no current should flow through the zener.

Voltage and Current monitors

The FDV304P has a maximum V_GS of 14.4V. Hence a fix resistor divider and the zener: Under normal 12V operation no current should flow through the zener.

PTC

PTC Resettable Fuses

500mA

Battery connectors

Terminal for external power supply

The FDV304P has a maximum V_GS of 14.4V. Hence a fix resistor divider and the zener: Under normal 12V operation no current should flow through the zener.

Voltage and Current monitors

The FDV304P has a maximum V_GS of 14.4V. Hence a fix resistor divider and the zener: Under normal 12V operation no current should flow through the zener.

PTC

PTC Resettable Fuses

500mA

Battery connectors

Terminal for external power supply

The FDV304P has a maximum V_GS of 14.4V. Hence a fix resistor divider and the zener: Under normal 12V operation no current should flow through the zener.

Voltage and Current monitors

The FDV304P has a maximum V_GS of 14.4V. Hence a fix resistor divider and the zener: Under normal 12V operation no current should flow through the zener.

PTC

PTC Resettable Fuses

500mA

Battery connectors

Terminal for external power supply

The FDV304P has a maximum V_GS of 14.4V. Hence a fix resistor divider and the zener: Under normal 12V operation no current should flow through the zener.

Voltage and Current monitors

The FDV304P has a maximum V_GS of 14.4V. Hence a fix resistor divider and the zener: Under normal 12V operation no current should flow through the zener.
Oscillator circuit and start up acceleration

Note that XTAL2 is on top and connected to the NFET.

The pull ups on the data lines prevent the card from entering a corrupted mode when the MNP2 is not driving its pins yet.
1-wire driver

Open drain simulation

Two pull-up pins are required because the MSP432 only has push-pull pins, while 1-wire mode is an open-drain pin.

SDI12_UC_IN should be connected to a pin in push-pull mode.

Redirectional I/O

I2C1_SCL 1-wire

The MSP432's UART uses positive logic, so the part can be used for down translation to VCC.

Internal silicon serial number

1-wire bus.

between internal and external serial numbers connected to the Internal serial number can be disabled. This allows to distinguish

Inversion and down translation

Up translation

SDI-12 uses negative logic. The input pins of the SN74LVC2G04 are 5.5V tollerant, so the part can be used for down translation to VCC.

Two UC pins are required because the

Open drain simulation

Two pull-up pins are required because the MSP432 only has push-pull pins, while 1-wire mode is an open-drain pin.

SDI12_UC_OUT should be connected to a pin in push-pull mode.

Internal silicon serial number

1-wire needs an open drain pin. MSP432 only has push-pull pins, while

1-wire needs an open drain pin. MSP432 only has push-pull pins, while
Prevent short circuit when both devices are driving the line.
Yes, those LEDs really do all need the same current limiting resistor. I was not just lazy. They are dimensioned for about $I_F = 2mA$ @ $V_F = 3.0V$. $I_{F\text{max}} = 20 mA$

Jumper to make sure that LEDs are not accidentally enabled when node is deployed.

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos

Muchos debug ledjos
All of these GPIO lines can be used as analog inputs. Hence they all have smoothing caps.
Max expected data voltage: 5.5V

SDI-12 Data

Note that this circuit does more than just ESD protection. It also makes sure that the SDI-12 lines are protected by the correct source resistance. See the SDI-12 standard specification for more information.

12V power rails protection

5V +/- 10% --> 5.5V

Note that this circuit does more than just ESD protection. It also makes sure that the SDI-12 lines are protected by the correct source resistance. See the SDI-12 standard specification for more information.

12V power rails protection

5V +/- 10% --> 5.5V

Note that this circuit does more than just ESD protection. It also makes sure that the SDI-12 lines are protected by the correct source resistance. See the SDI-12 standard specification for more information.

We might want to replace these TVS diodes with footprint compatible SMBJ series from Littlefuse. On first sight these seem to have steeper voltage curves.
Main Harting connector: Connects to Souriau plug.

BOLT header
Header to be replaced and reprinted once spec is received.