

|  |  |
| --- | --- |
| SolarBoatTwente\_SPS(I)  This document describes the element performance specifications of the SolarBoatTwente\_SPS (MPPT power sensor) and the SolarBoatTwente\_SPSI (interfaces with the SPS and CAN-Bus)    Author:   Niels Leijen  Reviewers:   Frank Somhorst  Sander Oosterveld  For Info:  Frank Somhorst |  |
|  |  |
|  |  |

Table of Contents

[Chapter 2 | Introduction 2](#_Toc3665695)

[1 Context 2](#_Toc3665696)

[Chapter 3 | Introduction 3](#_Toc3665697)

[1.1. Context 3](#_Toc3665698)

[Chapter 4 | RX CAN berichten 4](#_Toc3665699)

[Chapter 5 | TX CAN berichten 6](#_Toc3665700)

[Chapter 6 | Overig 7](#_Toc3665701)

**LIST OF FIGURES**

[Figure 3‑1 3](#_Toc3665702)

**LIST OF TABLES**

**No table of figures entries found.**

**DEFINITIONS**

|  |  |
| --- | --- |
| **CAN-Bus** | **Controller Area Network Bus** |
| **MPPT** | **Maximum Power Point Tracker** |
| **PV** | **Power Voltaics, more commonly known as solar panels** |
| **SBT** | **Solar Boat Twente** |
| **SPS** | **Solar Power Sensor** |
| **SPSI** | **Solar Power Sensor Interface** |
| **I2C** | **Inter IC (Bus)** |
|  |  |
|  |  |

# Introduction

This document describes the element performance specification of the SolarBoatTwente\_SolarPowerSensor (refered to as SBT\_SPS) and the SolarBoatTwente\_SolarPowerSensorInterface (refered to as SBS\_SPSI).

The SBT\_SPS will consist of a custom designed hardware element (Solar Power Meter) and sofware.  
The SBT\_SPSI will consist of a set of hardware elements (Arduino Rev3 and Sparkfun CAN-Bus Shield) and software.

The whole (SBT\_SPS and SBT\_SPSI) will be considered a ‘black box’ from the solar boat its perspective and relevant interfaces and protocols for communicating with the solar boat will be described in this document.

## Context

From the documentation delivered by Solar Boat Twente (in Dutch):

“Één van de cruciale onderdelen binnen de energiehuishouding in de zonneboot van Solar Boat Twente zijn de maximum power point trackers. De huidige MPPTs zijn ingekochte Genasuns en werken naar tevredenheid op één punt na: er kan niet uitgelezen worden hoeveel vermogen binnenstroomt van de zonnepanelen en doorstroomt naar de batterij. Hiervoor is een additionele PCB ontworpen welke batterij en paneel voltage kunnen meten, evenals batterij en paneel stromen. Daarnaast is er een relais ingebouwd welke de panelen aan/uit kunnen zetten. Binnen de zonneboot wordt data verstuurd over een CANbus. De additionele PCBs kunnen helaas niet direct op de CANbus aangesloten worden, er is gekozen voor een topologie waarbij alle additionele PCBS aan elkaar gekoppeld zijn middels een I2C verbinding welke vervolgens aangesloten moet worden op een andere microcontroller die de I2C data om kan zetten naar CAN. Voor deze interface welke de I2C data om kan zetten in CAN data wordt de hulp ingeschakeld van een oudgediende van SBT: Niels Leijen. Dit document beschrijft welke input/output data verwacht wordt bij deze I2C – CAN omzetting.  
De PCBs zijn zodanig ontworpen dat er in totaal 16 verschillende I2C adressen op geprogrammeerd kunnen worden middels een DIP-switch: adres 112 t/m 127. In de zonneboot zelf zijn vooralsnog slechts 10 panelen geïnstalleerd en worden er naar verwachting dus slechts 10 van deze adressen daadwerkelijk gebruikt. Het is de bedoeling dat er naar CAN adres 0x03C een commando door de boordcomputer/externe apparatuur gestuurd kan worden welke, afhankelijk van de meegestuurde data, ervoor zorgt dat relais aan/uit worden gezet. Verder is het de bedoeling dat met een frequentie van 1 CAN bericht per seconde de powerdata van elke PCB wordt doorgestuurd op de CANbus, welke uitgelezen kan worden door de boordcomputer voor verdere berekeningen ten behoeve van data acquisitie.”

In short; There is a need to measure the power of the solar panels and the current MPPTs are not able to measure this. The first SBT already made efforts to try to fix this issue but never achieved this due to a relative complex system to measure the currents (not in cited text). Therefore a new PCB was made which communicates via an I2C interface and can be easily programmed with the Arduino library. This PCB cannot directy interface with the CAN-Bus of the solar boat and needs an external interface to translate I2C commandos to CAN-Bus commandos.

SPSI

SPS  
1 … 16

CAN-Bus  
&  
+5V

I2C  
&  
+5V

Solar Power

Solar  
Boat

MPPT  
1 … 16

Battery

48V

Arduino Rev3

Sparkfun CAN-Bus

Figure 3‑1 Context diagram of the SBS and SPSI

## Scope

This document describes:

* The interfaces of the SPS and SPSI
* The software of the SPS and SPSI

## Functional Summary

### SPSI

#### CAN + Power In

The SPSI will communicate using the CAN-Bus. The same connection also supplies +5V to the SPSI via the power wires of the CAN bus and powers both the Arduino and the Sparkfun CAN-Bus board.

#### I2C + Power Out

To communicate with the SPS an I2C interface connection will be placed on the SPSI. This connection will also supply +5V power to the SPS.

### SPS

#### I2C + Power In/Out

The SPS will communicate via I2C connection, as well as receive its power via said connection. I2C + Power connection will be placed twice on the SPS in order to make daisy chaining up to 16 devices possible with the same I2C bus.

#### I2C ID

The I2C ID will be determined by 4 switches placed on the SPS

#### PV measurements

The PV current will be measured using a AD8418 and can be interrupted using a relay. The voltage will be measured directly by the microcontroller via a resistive divider connected in front of the relay.

#### Battery measurements

The current and voltage from the MPPT going to the battery is measured the same way as the PV. The only difference being that the power is not switched via the a relay.

## Interface Requirements

This chapter describes how the SPS and SPSI will respond on incoming CAN messages (in Dutch because the author of this document was lazy).

### CAN Samengevat

* CAN Baud rate used will be 250k
* TX CAN bericht voor ID=0x03C met aan/uit zetten van relais.
* RX CAN bericht voor een aantal IDs met vermogensdata. Automatisch uitzenden van een bericht met een interval van 1 seconde.

Het is de bedoeling dat I2C adres en CAN adres aan elkaar gekoppeld worden zoals weergegeven is in onderstaande tabel:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **I2C** | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |
| **CAN** | 0x03D | 0x03E | 0x03F | 0x040 | 0x041 | 0x042 | 0x043 | 0x044 | 0x045 | 0x046 | 0x047 | 0x048 | 0x049 | 0x04A | 0x04B | 0x04C |

#### RX CAN berichten

Er moet via CAN een bericht naar ID=0x03C gestuurd worden welke relais aan/uit kan zetten. De structuur van dit bericht is weergegeven zoals in onderstaande tabel:

RX CAN message structure description for ID=0x03C

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte[[1]](#footnote-1)** | **Beschrijving** | **Type** |  |
| 1 | Relais aan/uit high byte | Unsigned integer | Elke bit komt overeen met het aan/uitzetten van een bepaald relais van een bepaald paneel.  Bitwaarde:  0 – relais uit  1 – relais aan  LSB: I2C adres 112  MSB: I2C adres 127  Bijv. 1000 0100 0000 0010  🡪 Alle relais staan uit behalve die van de MPPT sensor met I2C adres 113, 122 en 127. |
| 2 | Relais aan/uit low byte |

#### TX CAN berichten

Er moet via CAN berichten uitgestuurd worden vanuit ID=0x03D t/m ID=0x04C met data over het vermogen wat in/uit de MPPT komt. Deze berichten moeten elke seconde automatisch verstuurd worden op de CANbus. De structuur van deze berichten zijn weergegeven zoals in onderstaande tabel:

TX CAN message structure description for ID=0x03D t/m ID=0x04C

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte** | **Beschrijving** | **Type** |  |
| 1 | VsensBat high byte | Unsigned integer | 0 – 1023 (ADC systeem meet slechts 1024 punten) |
| 2 | VsensBat low byte |
| 3 | IsensBat high byte | Unsigned integer | 0 – 1023 (ADC systeem meet slechts 1024 punten) |
| 4 | IsensBat low byte |
| 5 | VsensPV high byte | Unsigned integer | 0 – 1023 (ADC systeem meet slechts 1024 punten) |
| 6 | VsensPV low byte |
| 7 | IsensPV high byte | Unsigned integer | 0 – 1023 (ADC systeem meet slechts 1024 punten) |
| 8 | IsensPV low byte |

## Implementation Requirements

### Connectors

The pinout of the connectors used is described here.

Here could also be some sort of description of the connectors used (info about connector type, reference designator etc.)

#### SPSI

##### CAN

|  |  |
| --- | --- |
| Pin Nr. | X1 (on Sparkfun CAN-Bus) |
| 1 | NC |
| 2 | GND |
| 3 | CANH |
| 4 | NC |
| 5 | CANL |
| 6 | NC |
| 7 | NC |
| 8 | NC |
| 9 | Power In |

Other configurations possible via jumpers.

##### SPS

|  |  |
| --- | --- |
| Pin Nr. | (Connector not defined need custom wiring) |
| 1 | SDA |
| 2 | SCL |
| 3 | +5V |
| 4 | GND |

#### SPS

##### SPSI

|  |  |  |
| --- | --- | --- |
| Pin Nr. | P4 | P5 |
| 1 | GND | SDA |
| 2 | EX (+5V) | SCL |
| 3 | SCL | EX (+5V) |
| 4 | SDA | GND |

##### MPPT & Battery

|  |  |
| --- | --- |
| Pin Nr. | P1 |
| 1 | Battery Out |
| 2 | GND |
| 3 | GND |
| 4 | PV In |

## Functional and Performance Requirements

Here could be some info about what the current and/or voltage range of the mppt input/outputs.

Overig

De software welke momenteel draait op onze PCBs staat op Github. Deze is te bereiken via de volgende link: <https://github.com/SolarBoatTwente/SlavePowerSensor/blob/master/MPPT_test_sensor/MPPT_test_sensor.ino>

De hardware welke gebruikt wordt om de I2C – CAN verbinding te maken bestaat uit een Arduino UNO clone van Geekcreit met een CANbus shield van Sparkfun. Documentatie hiervoor is te bereiken via de volgende link:

<https://www.sparkfun.com/products/13262>

1. If 0 bytes are sent, the SPSI will rescan for SPS devices on the I2C bus [↑](#footnote-ref-1)