MatchX Dev Kit
LoRa Development Board with Grove Sensors
User Guide
Rev 1.0
1 Introduction ................................................................. 5
  1.1 Product overview .................................................... 5
    1.1.1 Lora ........................................................... 5
    1.1.2 BLE ........................................................... 6
  1.2 Main Features ......................................................... 6
    1.2.1 Hardware ...................................................... 6
    1.2.2 Software ...................................................... 6

2 Hardware Architecture - SoM module ............................... 7
  2.1 Pin-out and pin description of the SoM module .................. 7
    2.1.1 Dimensions .................................................. 9
  2.2 Operating frequency bands ......................................... 10
    2.2.1 EU 863-870MHz ISM Band .................................. 10
    2.2.2 US 902-928MHz ISM Band .................................. 10
    2.2.3 Australia 915-928MHz ISM Band ............................ 10

3 Hardware Architecture - Evaluation Board ......................... 13
  3.1 Block Diagram ...................................................... 13
  3.2 Hardware features ................................................. 15
  3.3 Connectors ......................................................... 16
    3.3.1 Connectors pin-out ......................................... 17
  3.4 Jumpers and test connectors ...................................... 20
3.5 Hardware selectable options ............................................. 21

4 Sensor connection ............................................................... 23
4.1 Grove Digital ................................................................. 23
4.2 Grove Analog ................................................................. 23
4.3 Grove UART ................................................................. 24
4.4 Grove I2C ................................................................. 24

5 Quick Installation Guide ...................................................... 25
5.1 Software and Hardware requirements ................................ 25
5.2 Connections ................................................................. 25
5.2.1 Power ................................................................. 25
5.2.2 Bluetooth connection .............................................. 26
5.3 Setup ................................................................. 26
5.4 Registering a node on MarchX server ................................ 26
5.5 Setting DevEUI, AppEUI and DevKey .............................. 27
5.6 Region Selection .............................................................. 28

6 Software Development Guide ............................................... 30
6.1 References ................................................................. 30
6.2 Prerequisites ................................................................. 30
6.3 Software development under Windows OS ....................... 31
6.3.1 Using SmartSnippet Studio ...................................... 33
6.4 Software development under Linux ................................ 36

7 Product specification .......................................................... 37
7.1 Software environment ..................................................... 37
7.2 Hardware environment .................................................. 37
7.2.1 RF performance .................................................... 38
7.2.2 Electrical characteristics ......................................... 38
7.2.3 Antenna characteristics ........................................... 39
7.3 Dimensions ................................................................. 40
7.4 Certification ............................................................... 41
1. Introduction

1.1 Product overview

The LPWAN Dev Kit by MatchX is a high performance, ready to use development platform allowing you to kick-start your IoT project. Together with a MatchX Core module the Dev Kit is an incredibly flexible solution that can be deployed in a various number of applications which require long distance communication and long battery life. The unique combination of both LoRa and Bluetooth Low Energy makes non-contact firmware updates easy, especially when the device is mounted in a unaccessible place.

This guide covers both the US and EU version of the Dev Kit. The main differences between these two versions are listed in Table 1.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>US</th>
<th>EU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Frequency Band</td>
<td>902-928MHz</td>
<td>863-870MHz</td>
</tr>
<tr>
<td>Maximum Output Power</td>
<td>+17dBm</td>
<td>+14dBm</td>
</tr>
<tr>
<td>Lora BW</td>
<td>500k/125kHz</td>
<td>125kHz</td>
</tr>
<tr>
<td>SF</td>
<td>7-10</td>
<td>7-12</td>
</tr>
<tr>
<td>Certification</td>
<td>IEC 60950-1</td>
<td>EN 300200</td>
</tr>
<tr>
<td></td>
<td>FCC PART 15.247</td>
<td>EN 301489</td>
</tr>
</tbody>
</table>

Table 1.1: Comparison of different regions

1.1.1 Lora

The MatchX Module uses LoRa communication to send messages over long distances (up to 20km in open spaces). This unique modulation scheme guarantees robust wireless communication even in difficult from RF point of view environments such as high-rise city landscapes or within the inside of buildings. The module can output up to 17dBm of power and is fully LoraWAN compatible. It’s uniquely designed to work with the MatchX Box gateway and can also be used with a LoraWAN
compatible Gateway of your choice.

1.1.2 BLE

The module offers a novel firmware solution upgrade by augmenting LoRa, together with Bluetooth Low Energy (BLE). As LoRa protocol is not suitable for transmitting large amounts of data, MatchX has combated this with BLE, offering a quick, robust and remote way of updating your software. It is a perfect method in cases where a sensor may be mounted in an unaccessible place like in a basement, sealed container box or behind a wall. Moreover BLE together with provided mobile app enables you to configure your module and read its status and additional data.

1.2 Main Features

LPWAN Dev Kits long range, long battery life, flexible sensor configuration and wireless firmware update are the key features that are offered by the Core module.

1.2.1 Hardware

- integrated MatchX Core System on Module
- +18.5dBm output power in 868MHz/915MHz
- -146dBm sensitivity of LoraWAN packets
- integrated Semtech SX1276 LoRa and Bluetooth Low Energy
- 0 Hz up to 96 MHz 32-bit ARM Cortex-M0 Dialog DA14680 microcontroller
- GPS receiver with 22 tracking / 66 acquisition-channels
- optional NB-IOT, Tri-Band LTE-FDD and Dual-Band GPRS/EDGE module
- integrated Li-ION battery charger
- 16 bit I/O expander
- GROVE sensor connector
- RGB indicator LED and User Button
- temperature sensor with 0.125°C resolution and 1°C accuracy
- unique ID EEPROM memory
- ultra low power design

1.2.2 Software

- Runs LoRa and Bluetooth stack simultaneously
- Low power consumption modes
- Easy to use software package
- Eclipse-based IDE
- Firmware upgrade over the air
- Mobile application

Currently there is no Class B support in server yet, but the hardware and firmware are fully prepared for Class B specification, it is expected to support Class B in future firmware upgrade.
2. Hardware Architecture - SoM module

2.1 Pin-out and pin description of the SoM module

The pin-out of the MatchX Core SoM module can be seen on Figure 2.1 and the description of the pins in Table 2.1. On top of the module there are two UFL RF connectors, the one on the left is the LoRa antenna connector, a suitable 868MHz in EU and 915MHz in US, 50 Ohm antenna is expected to be connected on these port. The other connector is for connecting the 2.4GHz, 50 Ohm BLE antenna. Both antennas come together with the evaluation board.

![Figure 2.1: Pin-out of the SoM module.](image-url)
The module can be powered in two ways:

1. By connecting the VBATT to a battery voltage (2.7V to 4.2V).
2. By supplying +5V on the VBUS pin.

If both power sources are present, the battery will be charged form +5V power supply. The charging current and charging characteristics for different battery types is software configurable. The module provides V3P3_LDO voltage, it is a output of internal LDO of the DA14680 MCU, and it can be used to supply external devices, but the maximum current drawn can’t be greater than 100mA. By default VDD_RFS is connected to V3P3_LDO with an external 0R resistor. The current draw of VDD_RFS is around 35mA during transmission with +14dBm power output and around 90mA with +17dBm power. This has to be taken in consideration when planning the power budget of V3P3_LDO. When even higher RF transmission power is required it is advisable to use different power source for VDD_RFS. On the Evaluation Board it can be done by using 3.3V output of the low power converter.

The source of V3P3_LDO is VBUS when present or VBATT otherwise. As it is a output of a LDO, when VBUS is not present, and VBATT drops below 3.3V the V3P3_LDO will follow the battery voltage. By default all GPIO are referenced to V3P3_LDO (it is also possible to configure 1.8V as the GPIO level, each GPIO can be configured individually) so care must be taken to ensure

<table>
<thead>
<tr>
<th>Pin number</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V3P3_LDO</td>
<td>3.3V output of the internal LDO</td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>3</td>
<td>VDD_RFS</td>
<td>Supply voltage of the radio front-end</td>
</tr>
<tr>
<td>4</td>
<td>LED1</td>
<td>Open drain output type, LED driver</td>
</tr>
<tr>
<td>5</td>
<td>LED2</td>
<td>Open drain output type, LED driver</td>
</tr>
<tr>
<td>6</td>
<td>LED3</td>
<td>Open drain output type, LED driver</td>
</tr>
<tr>
<td>7</td>
<td>RESET</td>
<td>Reset signal, active high</td>
</tr>
<tr>
<td>8</td>
<td>P1_6</td>
<td>General Purpose I/O P1_6 / NTC resistor for battery temperature sensing</td>
</tr>
<tr>
<td>9</td>
<td>P1_4</td>
<td>General Purpose I/O P1_4 / ADC1 / battery temperature sensing</td>
</tr>
<tr>
<td>10</td>
<td>P4_2</td>
<td>General Purpose I/O P4_2</td>
</tr>
<tr>
<td>11</td>
<td>P4_3</td>
<td>General Purpose I/O P4_3</td>
</tr>
<tr>
<td>12</td>
<td>P2_3</td>
<td>General Purpose I/O P2_3</td>
</tr>
<tr>
<td>13</td>
<td>P1_3</td>
<td>General Purpose I/O P1_3 / ADC2</td>
</tr>
<tr>
<td>14</td>
<td>P0_7</td>
<td>General Purpose I/O P0_7 / ADC3</td>
</tr>
<tr>
<td>15</td>
<td>SWD_DIO</td>
<td>Serial Wire Debug interface I/O signal / GPIO P0_6 / ADC4</td>
</tr>
<tr>
<td>16</td>
<td>SWD_CLK</td>
<td>Serial Wire Debug interface clock signal / GPIO P2_4 / ADC7</td>
</tr>
<tr>
<td>17</td>
<td>P3_3</td>
<td>General Purpose I/O P3_3</td>
</tr>
<tr>
<td>18</td>
<td>P3_4</td>
<td>General Purpose I/O P3_4</td>
</tr>
<tr>
<td>19</td>
<td>P3_2</td>
<td>General Purpose I/O P3_2</td>
</tr>
<tr>
<td>20</td>
<td>VBATT</td>
<td>Battery voltage input</td>
</tr>
<tr>
<td>21</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>22</td>
<td>VBUS</td>
<td>5V supply, charging voltage</td>
</tr>
</tbody>
</table>

Table 2.1: USB-C connector pins description.
2.1 Pin-out and pin description of the SoM module

that no voltage higher than \textbf{V3P3\_LDO} is presented to any GPIO. This may happen when powering external devices, that connect to SoM module, from a boost converter.

![Block diagram of the Core module.](image)

\textbf{Figure 2.2:} Block diagram of the Core module.

\textbf{2.1.1 Dimensions}

![Dimension of the SoM module.](image)

\textbf{Figure 2.3:} Dimension of the SoM module.
2.2 Operating frequency bands

2.2.1 EU 863-870MHz ISM Band

In the European region the EN300220-2 V3.1.1 (2017-02) regulation defines the allowed frequency allocation and spectrum access. Every device working in this band must comply with these rules as shown in the Table 2.3. EU regulations restrict the maximum radiated power as well as the duty cycle of the transmission in different frequency bands. To comply with the duty cycle requirement the transmitting device must wait after every transmitted packet. The time device has to wait depends on the time on air of transmitted packet and this in turn depends on the length of the packet and spreading factor SF. This relation and required wait time can be seen in Table 2.2 According to LoRaWAN specification every device has to implement at least 3 channels as follows:

- 868.10 MHz
- 868.30 MHz
- 868.50 MHz

The SoM is preconfigured to work with the MatchX Box gateway and additionally to the 3 mandatory channels 5 additional channels are defined. The list of all preconfigured channels can be found in Table 2.4.

<table>
<thead>
<tr>
<th>Spreading Factor (125kHz Lora)</th>
<th>Bit rate (bps)</th>
<th>Range (depends on conditions)</th>
<th>Time on air (ms) (10 bytes payload)</th>
<th>0.1% duty cycle waiting time</th>
<th>1% duty cycle waiting time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF7</td>
<td>5470</td>
<td>2 km</td>
<td>56 ms</td>
<td>1 min</td>
<td>6s</td>
</tr>
<tr>
<td>SF8</td>
<td>3125</td>
<td>4 km</td>
<td>100 ms</td>
<td>1 min 40s</td>
<td>10s</td>
</tr>
<tr>
<td>SF9</td>
<td>1760</td>
<td>6 km</td>
<td>200 ms</td>
<td>3 min 20s</td>
<td>20s</td>
</tr>
<tr>
<td>SF10</td>
<td>980</td>
<td>8 km</td>
<td>370 ms</td>
<td>6 min 10s</td>
<td>37s</td>
</tr>
<tr>
<td>SF11</td>
<td>440</td>
<td>14 km</td>
<td>740 ms</td>
<td>12 min 20s</td>
<td>1 min 14s</td>
</tr>
<tr>
<td>SF12</td>
<td>290</td>
<td>20 km</td>
<td>1400 ms</td>
<td>23 min 20s</td>
<td>2 min 20s</td>
</tr>
</tbody>
</table>

Table 2.2: Modules operating frequencies.

2.2.2 US 902-928MHz ISM Band

These frequencies band can be used in USA, Canada and all other countries that adopt the entire FCC-Part15 regulations in 902-928 ISM band. For these region MatchX uses predefined frequencies listed in Table 2.5. The FCC regulation puts restriction on the maximum dwell time of 400ms in uplink, thats why the maximum allowed spreading factor is SF10.

2.2.3 Australia 915-928MHz ISM Band

These frequencies band can be used in Australia region. For these region MatchX uses predefined frequencies listed in Table 2.6. All channels use 125kHz bandwidth and maximum of +20dBm output power can be reached.
### 2.2 Operating frequency bands

<table>
<thead>
<tr>
<th>Operational Frequency band</th>
<th>Maximum e.r.p</th>
<th>Channel access and occupation rules (e.g. Duty cycle or LBT + AFA)</th>
<th>Band number from EC Decision 2013/752/EU [i.3]</th>
<th>Class 1 subclass number according Commission Decision 2000/299/EU [i.7]</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>863,000 MHz to 865,000 MHz</td>
<td>25 mW e.r.p.</td>
<td>≤ 0,1% duty cycle or polite spectrum access</td>
<td>46a 66</td>
</tr>
<tr>
<td>L</td>
<td>865,000 MHz to 868,000 MHz</td>
<td>25 mW e.r.p.</td>
<td>≤ 1% duty cycle or polite spectrum access</td>
<td>47 67</td>
</tr>
<tr>
<td>M</td>
<td>868,000 MHz to 868,600 MHz</td>
<td>25 mW e.r.p.</td>
<td>≤1% duty cycle or polite spectrum access</td>
<td>48 28</td>
</tr>
<tr>
<td>N</td>
<td>868,700 MHz to 869,200 MHz</td>
<td>25 mW e.r.p.</td>
<td>≤0,1% duty cycle or polite spectrum access</td>
<td>50 29</td>
</tr>
<tr>
<td>O</td>
<td>869,400 MHz to 869,650 MHz</td>
<td>25 mW e.r.p.</td>
<td>≤0,1% duty cycle or polite spectrum access</td>
<td>54a 130</td>
</tr>
<tr>
<td>P</td>
<td>869,400 MHz to 869,650 MHz</td>
<td>500 mW e.r.p.</td>
<td>≤10 % duty cycle or polite spectrum access</td>
<td>54b 30</td>
</tr>
<tr>
<td>Q</td>
<td>869,700 MHz to 870,000 MHz</td>
<td>5 mW e.r.p.</td>
<td>No requirement</td>
<td>56a 31</td>
</tr>
<tr>
<td>R</td>
<td>869,700 MHz to 870,000 MHz</td>
<td>25 mW e.r.p.</td>
<td>≤1% duty cycle or polite spectrum access</td>
<td>56c 69</td>
</tr>
</tbody>
</table>

Table 2.3: EU wide harmonized national radio interfaces.

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Bandwidth</th>
<th>Maximum e.r.p</th>
<th>Channel access</th>
</tr>
</thead>
<tbody>
<tr>
<td>864.7 MHz</td>
<td>125 kHz</td>
<td>14 dBm</td>
<td>≤0,1% duty cycle</td>
</tr>
<tr>
<td>864.9 MHz</td>
<td>125 kHz</td>
<td>14 dBm</td>
<td>≤0,1% duty cycle</td>
</tr>
<tr>
<td>865.1 MHz</td>
<td>125 kHz</td>
<td>-4.5 dBm</td>
<td>≤1% duty cycle</td>
</tr>
<tr>
<td>865.3 MHz</td>
<td>125 kHz</td>
<td>-4.5 dBm</td>
<td>≤1% duty cycle</td>
</tr>
<tr>
<td>868.1 MHz</td>
<td>125 kHz</td>
<td>14 dBm</td>
<td>≤1% duty cycle</td>
</tr>
<tr>
<td>868.3 MHz</td>
<td>125 kHz</td>
<td>14 dBm</td>
<td>≤1% duty cycle</td>
</tr>
<tr>
<td>868.5 MHz</td>
<td>125 kHz</td>
<td>14 dBm</td>
<td>≤1% duty cycle</td>
</tr>
<tr>
<td>868.8 MHz</td>
<td>125 kHz</td>
<td>14 dBm</td>
<td>≤0,1% duty cycle</td>
</tr>
</tbody>
</table>

Table 2.4: Core Module operating frequencies in EU 863-870MHz ISM Band.
### Table 2.5: Modules operating frequencies (uplink) in US 902-928MHz ISM Band.

<table>
<thead>
<tr>
<th>Channel number</th>
<th>Frequency</th>
<th>Channel number</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>903.90 MHz</td>
<td>5</td>
<td>904.70 MHz</td>
</tr>
<tr>
<td>2</td>
<td>904.10 MHz</td>
<td>6</td>
<td>904.90 MHz</td>
</tr>
<tr>
<td>3</td>
<td>904.30 MHz</td>
<td>7</td>
<td>905.10 MHz</td>
</tr>
<tr>
<td>4</td>
<td>904.50 MHz</td>
<td>8</td>
<td>905.30 MHz</td>
</tr>
</tbody>
</table>

### Table 2.6: Modules operating frequencies (uplink) in Australia 915-928MHz ISM Band.

<table>
<thead>
<tr>
<th>Channel number</th>
<th>Frequency</th>
<th>Channel number</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>915.20 MHz</td>
<td>5</td>
<td>916.00 MHz</td>
</tr>
<tr>
<td>2</td>
<td>915.40 MHz</td>
<td>6</td>
<td>916.20 MHz</td>
</tr>
<tr>
<td>3</td>
<td>915.60 MHz</td>
<td>7</td>
<td>916.40 MHz</td>
</tr>
<tr>
<td>4</td>
<td>915.80 MHz</td>
<td>8</td>
<td>916.60 MHz</td>
</tr>
</tbody>
</table>
3. Hardware Architecture - Evaluation Board

3.1 Block Diagram

The development board comprises of two main subsystems:
1. MatchX SoM with LoRa and Bluetooth radios
2. SIMCom SIM7000E NB-IOT module with 3G/4G and GPS

Figure 3.1: Block diagram of the Dev Kit.
Chapter 3. Hardware Architecture - Evaluation Board

MatchX SoM is the core of the Dev Kit. It is responsible for controlling the SIMCom module, accessing the sensors, I/O expender, RGB LED etc. It also controls different power rails enabling low power modes.

The Dev Kit can be powered by USB-C +5V and/or Lithium-ion battery connected to X401 connector. The presence of +5V is signaled by a red LED next to USB-C. When both +5V and battery are present the MatchX SoM will start to charge the battery. The battery type, charging current, and charging curve are software configurable.

MatchX SoM provides V3P3_LDO output, which is a low current output of internal LDO capable to deliver up to 100mA of current. As it is used to power the LoRa RF front-end by default, it can be used to power just very low power peripherals. If more current is needed, an onboard SMPS TPS62740 should be used. It outputs a 3.3V up to 300mA current. Its output should be connected to the V3P3 power rail with the J402 jumper in position 1-2.

The SIMCom module is optional and is not included in a standard package but all peripheral components are already soldered. The main communication interface between MatchX SoM and SIMCom module is UART together with two control lines (SIM_NRST and SIM_PWR_KEY). Please refer to SIMCom SIM7000E datasheet to find more information about the modules operation.

The SIM7000E is powered from +5V provided by USB-C connector which is converted to 3.3V by a LDO. Its enable pin can be controlled by a SoM or can be always set high by a jumper.

The Dev Kit offers two ways to receive GPS signal. One is by using aforementioned SIM7000E module. A cheaper and less power demanding alternative is using SIM28ML.
3.2 Hardware features

Components marked on the Figure 3.2 are:

- Optional SIMCOM SIM7000E NB-IOT module with 3G/4G and GPS
- MatchX SoM module with LoRa and Bluetooth
- SIM28ML GPS receiver
- 3.3V low power converter for sensors supply
- PCA6416A I2C I/O expander
- 3.3V LDO for powering the optional SIM7000E module
- RGB LED controlled by the MatchX SoM
- User button, connected to P3_2 of the SoM module
- Reset button for MatchX SoM
- PCT2075GV I2C temperature sensor
Chapter 3. Hardware Architecture - Evaluation Board

I2C Serial EEPROM 24AA025UIDT-I/OT
- LED indicating presence of USB +5V
- NET indication LED of the SIM7000E
- LED indicating presence of SIM card

3.3 Connectors

![Connectors on the Dev Kit.](image)

Connectors marked on the Figure 3.3 are:

- J102 MatchX SoM modules I/Os
- J201 SIM7000E I/Os and signals
- J103 PCA6416A I/Os
3.3 Connectors

- J100 RGB LED control signals
- J301 Grove sensors I2C connector
- J101 MatchX SoM SWD programming and debugging connector
- J300 I2C connector
- X401 S2B-PH-SM4-TB Lithium battery connector
- USB-C connector, +5V supply and programming
- SIM card socket for SIM7000E

3.3.1 Connectors pin-out

Figure 3.4: Pin-out of the Dev Kit.

The J102 connector, shown on Figure 3.5 routes out all GPIOs available on the MatchX SoM module. Some of these GPIOs are used by default to control functions of the Dev Kit.
Chapter 3. Hardware Architecture - Evaluation Board

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P2_3</td>
<td>UART_SIM_RX</td>
<td>UART interface to SIMCom module, connected with J109 jumper</td>
</tr>
<tr>
<td>2</td>
<td>P1_3</td>
<td>UART_SIM_TX</td>
<td>UART interface to SIMCom module, connected with J105 jumper</td>
</tr>
<tr>
<td>3</td>
<td>P4_3</td>
<td>I2C_SCL</td>
<td>Main I2C bus, connected with J104 jumper</td>
</tr>
<tr>
<td>4</td>
<td>P4_2</td>
<td>I2C_SDA</td>
<td>Main I2C bus, connected with J108 jumper</td>
</tr>
<tr>
<td>5</td>
<td>P1_4</td>
<td>-</td>
<td>Not used</td>
</tr>
<tr>
<td>6</td>
<td>P1_6</td>
<td>-</td>
<td>Not used</td>
</tr>
<tr>
<td>7</td>
<td>P0_7</td>
<td>-</td>
<td>Not used</td>
</tr>
<tr>
<td>8</td>
<td>P3_3</td>
<td>-</td>
<td>Not used</td>
</tr>
<tr>
<td>9</td>
<td>P3_4</td>
<td>PS_EN</td>
<td>controls enable pin of TPS62740</td>
</tr>
<tr>
<td>10</td>
<td>P3_2</td>
<td>USR_BUTTON</td>
<td>User button connection</td>
</tr>
<tr>
<td>11</td>
<td>V3P3</td>
<td>3.3V</td>
<td>3.3V power, the source selectable by J402 jumper</td>
</tr>
<tr>
<td>12</td>
<td>GND</td>
<td>GND</td>
<td>Ground</td>
</tr>
</tbody>
</table>

Table 3.1: Functions assignment of the MatchX SoM GPIOs

Figure 3.5: Pin-out of J102 connector.

Figure 3.6: Pin-out of J301 connector.
3.3 Connectors

The J103 connector, shown on Figure 3.7, exposes all pins available on the PCA6416A I2C I/O expander. It offers 16 I/Os organized in two ports P0 and P1. Each pin can be configured individually as a input or output, and its state can be read and set by the I2C commands. Additionally there is a interrupt line EXP_INT that is being driven by the PCA6416A when input IO changes its state.

![Figure 3.7: Pin-out of J103 connector.](image)

The J201 connector, depicted on Figure 3.8, exposes all unused pins of the SIMCom SIM7000E module. For more information about the signals functions please refer to the modules datasheet.

![Figure 3.8: Pin-out of J201 connector.](image)
3.4 Jumpers and test connectors

Components marked on the Figure 3.9 are:

- **J402** - selection of the source of V3P3 used to power RGB LED and sensors. A jumper in position 1-2 selects the 3V3 from low power converter (located left), jumper in position 2-3 selects the V3P3_LDO output from integrated LDO of the SoM module.
- **J111, J112** - connect the P1_3 and P0_7 to Grove D1 and D2 lines.
- **J104, J108** - connect the P4_2 and P4_3 to I2C_SDA and I2C_SCL.
- **J105, J109** - connect P1_2 and P2_3 of MatchX SoM to UART_SIM_TX and UART_SIM_RX of the SIM7000E module.
- **J107** - connects the enable line of the 3.3V LDO that powers SIM7000E ether to EXT_P0_2 (I/O expander) in position 1-2 or to 5V (always on) in position 2-3.
- **J110** - connects SIM_NRST of the SIM7000E to EXT_P0_1 of the I/O expander.
3.5 Hardware selectable options

- **J106** - connects SIM_PWR_KEY of the SIM7000E to EXT_P0_0 of the I/O expander
- **J404** - test connector of the battery voltage
- **J401** - connects USB +5V to the input of the LDO, it is bypassed by a 0R resistor by default
- **J403** - connects power to the input of the low power 3.3V converter, the source of the power is USB +5V or battery if the +5V is not present.
- **J400** - test connector of the VCC_SIM (by default it is a 3.3V output of the LDO)

---

Figure 3.10: Hardware options of the Dev Kit.

Components marked on the Figure 3.10 are:

- 0R jumper connects P3_2 of SoM module to the User Button
0R jumper connects P3_4 of SoM module to PS_EN, a enable signal of the low power 3.3V converter

OR jumper selectin the source of VDD_RFS (the radio frontend power). The OR connected on the left side connects the VDD_RFS to low power 3.3V converter, jumper soldered on the right side connects VDD_RFS to the V3P3_LDO (the output of internal LDO of the SoM)

2.2M Ohm resistor connected in the upper position pulls the enable line of the low power 3.3V converter low, in the lower position pulls this line up

test point connected to the interrupt line of the temperature sensor (PTC_OS)

test point connected to the power good pin of low power 3.3V converter

test point connected to the CNTRL pin of low power 3.3V converter, pulling it high enables the internal LDO

test point connected to the LOAD pin of low power 3.3V converter, it is a output of internal LDO

test point connected to the SIM_MDM_LOG pin of the SIM7000E module

test point connected to the SIM_BOOT_CFG pin of the SIM7000E module

OR jumpers connect P2_3 and P3_3 GPS_UART_RXD and GPS_UART_TXD, they need to be removed when SIM7000 is to be used
4. Sensor connection

The Dev Kit is specifically designed to work with all different sorts of sensors, which can be attached to its GROVE connector or to 2.54mm pin headers. This makes it a very flexible solution that can be adjusted to specific applications and individual needs.

4.1 Grove Digital

The Grove digital modules use two signal lines called D0 and D1. There are, for example, switch Modules, the Fan Module, and the LED Module that controlled by digital GPIOs D0 and D1.

The following table shows the pin out of the Grove digital connector:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>D0</td>
<td>Primary Digital I/O</td>
</tr>
<tr>
<td>2</td>
<td>D1</td>
<td>Secondary Digital I/O</td>
</tr>
<tr>
<td>3</td>
<td>VCC</td>
<td>Power for Grove module</td>
</tr>
<tr>
<td>4</td>
<td>GND</td>
<td>Ground for Grove module</td>
</tr>
</tbody>
</table>

Table 4.1: Pin-out of the Grove digital connector

4.2 Grove Analog

The Grove Analog modules use two signal lines called A0 and A1. There are, for example, Potentiometer, Voltage Divider, and the Air Quality Module that controlled by analog pins A0 and A1.

The following table 4.2 shows the pin out of the Grove Analog connector:
Chapter 4. Sensor connection

### 4.3 Grove UART

The Grove UART modules use two signal lines RX and TX. There are, for example, RFID module that controlled by UART pins TX and RX.

The following table 4.3 shows the pin out of the Grove UART connector:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RX</td>
<td>Serial receive</td>
</tr>
<tr>
<td>2</td>
<td>TX</td>
<td>Serial transmit</td>
</tr>
<tr>
<td>3</td>
<td>VCC</td>
<td>Power for Grove module</td>
</tr>
<tr>
<td>4</td>
<td>GND</td>
<td>Ground for Grove module</td>
</tr>
</tbody>
</table>

Table 4.3: Pin-out of the Grove UART connector

### 4.4 Grove I2C

The Grove I2C modules use two signal lines SDA and SCL. There are, for example, motion detect module and environment module that controlled by I2C pins SDA and SCL.

The following table 4.4 shows the pin out of the Grove I2C connector:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SCL</td>
<td>I2C Clock</td>
</tr>
<tr>
<td>2</td>
<td>SDA</td>
<td>I2C data</td>
</tr>
<tr>
<td>3</td>
<td>VCC</td>
<td>Power for Grove module</td>
</tr>
<tr>
<td>4</td>
<td>GND</td>
<td>Ground for Grove module</td>
</tr>
</tbody>
</table>

Table 4.4: Pin-out of the Grove UART connector
5. Quick Installation Guide

The Core module incorporated in the Dev Kit is by default configured to connect with MatchX LoraWAN cloud server if the network coverage is present. Nonetheless some settings can be changed and some set up may be required in order to use the Dev Kit in a custom application. This installation guide will introduce how to set up the LoRa parameters and establish LoraWAN network connection.

5.1 Software and Hardware requirements
To configure the board, users will need to have the following:

- Dev Kit with USB-C power cable or Li-ion battery
- MatchX server account
- LoraWAN coverage by e.g. MatchX Box gateway
- USB-to-UART converter
- PC computer with Windows or Linux OS

5.2 Connections
5.2.1 Power
The Dev Kit is equipped with battery connector. As long as the battery is charged there is no other connection required for the Dev Kit to work. Charging: If the LED is signaling that the battery needs to be charged or the state of charge readouts from the mobile app or through LoRa are indicating low battery it can be charged two ways:

1. By connecting the Dev Kit to a USB charger. A standard USB charger with 5V output and minimum 500mA of current can be used and standard, reversible USB-C cable. The integrated charging controller will take care of proper charging of the battery. This process can take up to 12h.
2. By using a dedicated lithium battery charger. Using this method may result in a faster charging process due to higher charging current. Please refer to the battery datasheet for details about the charging characteristics.
Chapter 5. Quick Installation Guide

Important!
MatchX strongly recommends to charge the battery within specified temperature range of 0 to +45°C. Charging outside of these recommended conditions may lead to either reduced battery life or permanent damage.

5.2.2 Bluetooth connection

The Core module implements Bluetooth Low Energy (Bluetooth 4.2 specification) with SUOTA (Software Update Over The Air) feature. The only hardware requirement is a connection of 2,4GHz, 50 Ohm antenna to BLE antenna port.

5.3 Setup

Every Dev Kit and Core module comes with preprogrammed unique MAC address, AppEUI and LoraWAN DevKey (also referred as AppKey by different sources). The DevKey is used to ensure a secure communication and data encryption between the module and application server. AppEUI is used to communicate with the application that registered on the Lora server. Care must be taken with storing the DevKey in a safe place and ensuring it is not compromised.

The products will come with a QR code sticker, which gives the Serial Number of the device. By typing in the S/N at the registration of the MatchX LPWAN Cloud, the preprogrammed APP EUI, MAC address and DevKey will be associated automatically.

5.4 Registering a node on MatchX server

Registering a node on MatchX server is a straightforward process. The user needs to know nodes DevKey, DevEUI and AppEUI. For more information about these keys refer to section 5.5.

Go to matchx.io and under 'Cloud' find an appropriate server according to region the node should be deployed. In this example we are using https://eux.matchx.io for Europa region. Register an account using valid email address. Go to your dashboard and click on 'Application' tab and than press 'Create application' button (see Figure 5.1).

![Figure 5.1: Creating new Application.](image)

Fill out the 'Application name' and 'Application description' fields and click 'Submit'. Click on your newly created application and under 'Nodes' tab click on the 'Create node' button.

Fill out information about DevKey, DevEUI and AppEUI. Device EUI should be in 64bit format (with fffe in the middle) like on Figure 5.2. Click on 'Submit' button. The node is now created, you
5.5 Setting DevEUI, AppEUI and DevKey

To ensure the highest level of security in LoRaWAN network and Over the Air Activation (OTTA) 3 different keys have to be programmed into every end node.

- **DevEUI** - 6 bytes global end-device ID in IEEE EUI48 address space that uniquely identifies the end-device, also used by Bluetooth. It is converted to IEEE EUI64 by inserting 0xFFFE in the bytes 4 and 5. e.g. 78af58fffe040000
- **AppEUI** - 8 bytes global application ID in IEEE EUI64 address space that uniquely identifies the application provider (i.e., owner) of the end-device

---

**Figure 5.2: Creating new node.**

You can click on 'View' under 'Frame Logs' to see all messages belonging to the node like it is shown on Figure 5.3.

**Figure 5.3: Nodes messages.**
Chapter 5. Quick Installation Guide

- **DevKey** - 16 bytes unique AES-128 key

These keys are programmed by MatchX and stored in a special region of the nonvolatile memory of the Dialog microcontroller. They will be preserved during flashing of the new firmware, however they will be lost by performing full flash erase. The default values are defined in `lora\param.c`. In Dev Kit Firmware the values of these keys are printed on UART console on power up, see Figure 5.4. On default DKF configures pins 5 and 6 on the J102 connector to be UART TX and RX respectively, a UART-to-USB converter needs to be connected to these pins to see the messages.

![Figure 5.4: Displaying the keys on UART console.](image)

The values of DevEUI, AppEUI and DevKey can also be changed from UART console by using `param` command. The syntax is as follow:

```
param x value
```

where \( x = 0 \) for DevEUI, \( 1 \) for AppEUI and \( 2 \) for DevKey, value is a hexadecimal value to be set.

When value field is not specified the command will output current value of the parameter (except DevKey, which will not be shown).

DevKey is a AES-128 encryption key used for secure communication. It should be kept secret and only known to the sensor owner, this is why it is recommended to not display it on the UART in a final version of the firmware by removing `#define DEBUG` line in `param.c` file. It is also recommended to change its value before registering the node on the server.

### 5.6 Region Selection

Dev Kit Firmware implements different regional settings according to regulations applicable to these regions. By default the board determines which settings to choose by reading the jumpers configuration on connector J103 during startup. Jumpers placement is shown on Figure 5.5 and configuration settings in Table 5.1.

The jumper configuration is read only if the settings are not programmed to internal memory of the module. It is possible to program these settings using UART similarly to programming the DevEUI in paragraph 5.5. In this case the jumper values are ignored and J103 pins can be used as regular GPIOs. The syntax is as follow:
param x value

where \( x = 5 \) for changing the region, value is a region number in hexadecimal (see Table 5.1). If value parameter is set to FF firmware will unset regional parameter and the board will return to reading jumpers settings.

![Figure 5.5: Jumpers placement for region selection (EU region selected).](image)

<table>
<thead>
<tr>
<th>Region number</th>
<th>Jumper Position</th>
<th>Region</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0 0 0 0 0</td>
<td>EU</td>
<td>Europe region</td>
</tr>
<tr>
<td>01</td>
<td>0 0 0 1 0</td>
<td>AS1</td>
<td>Asia AS923MHz ISM Band</td>
</tr>
<tr>
<td>02</td>
<td>0 0 1 0 0</td>
<td>KR</td>
<td>Korea region</td>
</tr>
<tr>
<td>08</td>
<td>0 1 0 0 0</td>
<td>US</td>
<td>USA region, 8 channels</td>
</tr>
<tr>
<td>09</td>
<td>0 1 0 1 0</td>
<td>AU</td>
<td>Australia region, 8 channels</td>
</tr>
<tr>
<td>18</td>
<td>1 1 0 0 0</td>
<td>US(full)</td>
<td>USA region, full 64 channels</td>
</tr>
<tr>
<td>19</td>
<td>1 1 0 1 0</td>
<td>AU(full)</td>
<td>Australia region, full 64 channels</td>
</tr>
</tbody>
</table>

Table 5.1: Region selection settings.

The purpose of this chapter is to help user to quickly install all necessary software components and establish hardware connections needed to start software development using MarchX Core SoM and Development Kit. MatchX is providing the Dev Kit Firmware (DKF) to be a starting point for further software development according to individual needs.

6.1 References

SoM module is based on Dialog DA14680 microcontroller so it is advisable to get familiar with the following documents available on Dialog Semiconductors website:

- DA14680-01 DS, Datasheet, Dialog Semiconductor
- UM-B-057-SmartSnippets Studio user guide, User manual, Dialog Semiconductor
- UM-B-047 DA1468x Getting Started, User manual, Dialog Semiconductor
- UM-B-044 DA1468x Software Platform Reference, User manual, Dialog Semiconductor
- UM-B-056 DA1468x Software Developer’s Guide, Dialog Semiconductor

6.2 Prerequisites

- MatchX Development Kit
- USB-C cable and 5V charger
- UART-USB converter
- JLink programmer
- Dialog’s Semiconductor SmartSnippets DA1468x SDK
- SmartSnippets Studio package
- MatchX Dev Kit Firmware
- Operating System (Windows or Linux)
6.3 Software development under Windows OS

The easiest way to install and configure all required tools is to install Dialogs SmartSnippets Studio package (it can be downloaded from the company website after registration). Experienced users can try to install all cross-compilation tools and configure their favorite SDE manually, but using Dialogs software the whole process is straightforward. Please follow UM-B-057 User Manual from Dialog for details about the installation.

After successful installation of SmartSnippets Studio and J-Link programmer, you should have gcc cross-compilation tools installed and proper PATH entry should exist, to check it you can open command line window and type `arm-none-eabi-gcc -v`. The result should be similar to what is shown on Figure 6.1. In these examples, we are using gcc version 4.9.3 20150529.

![Image of command prompt showing gcc version](image.png)

**Figure 6.1: Checking ARM tools installation.**

Download the Dialog’s Semiconductor SmartSnippets DA1468x SDK (in the example the SDK version 1.0.8.1050.1 has been used) and MatchX Dev Kit Firmware. Both SDK and Dev Kit Firmware should be put in one folder (for example SmartSnippet workspace folder). DKF folder contains a make file which can be executed by navigating to the firmware folder and typing `make` in the command line window. This command will compile the firmware. If everything has been set up correctly, the compilation process should return no errors and a binary file should be generated as a result, see Figure 6.2.

After the software has been successfully compiled, it can be programmed through J-Link programmer using a script provided by Dialog Semi. In command line window navigate to the DKF folder. The programming script `initial_flash.bat` should be located in SDK folder: `DA1468x_SDK_BTLE_v_1.0.8.1050.1\utilities\scripts\suota\v11\`

It takes two parameters - path to the .bin file with firmware and path to the J-Link tools. The syntax is as follows:

```
(Path)\initial_flash.bat "(Path to firmware)" "(Path to J-Link)"
```

The example of the command can be seen on Figure 6.3. Before executing it, the Dev Kit board has to be powered on and J-Link programmer has to be connected to SWD port on J101. Only GND, SWDCK and SWDIO are necessary to program the board. After successful programming process the screen should look similar as on Figure 6.4. On default DKF configures pins 5 and 6 on the J102 connector to be UART TX and RX respectively. By connecting a UART-to-USB converter to
these pins the firmware will output the console messages. The output information sent after reset and UART configuration can be seen on Figure 6.5.
6.3 Software development under Windows OS

6.3.1 Using SmartSnippet Studio

As MatchX DKF is a makefile based project it is possible to port it quite easily to different IDE and use different operating systems. SmartSnippet Studio is a Dialog Semiconductors IDE based on Eclipse. It offers makefile project import capabilities.

In order to import the project, open the SmartSnippet IDE. The folder structure should be the same as in previous section, both SDK and DKF should be in SmartSnippet workspace folder. Go to File->Import and choose 'Existing Code as Makefile Project' like on Figure 6.6.

![Figure 6.5: Console output of the Dev Kit after reset.](image)

![Figure 6.6: Import makefile project window.](image)

Click Next. On the next window navigate to the DKF folder. Choose 'Cross ARM GCC' and press Finish. The software should be correctly imported and you should be able to compile it by
going to **Project->Build All** or pressing the build icon.

To program the just compiled firmware into DK you need to import "scripts" project to your workspace. To do that go to **File->Import** and choose ‘**Existing Projects into Workspace**’ like on Figure 6.7.

![Figure 6.7: Import existing project window.](image)

Click **Next**. On the next window navigate to the script folder that should be located in `<sdk_root>\utilities\scripts`. See Figure 6.8 and Figure 6.9

![Figure 6.8: Importing eXisting project browse window.](image)

Now all the scripts should be available, but in order to use them they must be slightly modified to point to a correct .bin file. Click on ’**External Tools Configuration**’ as shown on Figure 6.10.
It is best to copy the 'suota_initial_flash_jtag_win' script by right clicking on it and pressing 'Duplicate' then renaming it. The 'Argument' section has to be modified to contain correct path to compiled firmware. The variables values can be modified by clicking 'Variables' button. The compiled .bin file is stored in obj folder in the project directory.
6.4 Software development under Linux

Software developing under Linux operating system is straight forward. The easiest way to setup the environment is to install the the SmartSnippet Studio from Dialog and following the installation guide in UM-B-057 User guide from Dialog. Download the Dialog’s Semiconductor SmartSnippets DA1468x SDK (in the example the SDK version 1.0.8.1050.1 has been used) and MatchX Dev Kit Firmware. Both SDK and Dev Kit Firmware should be put in one folder (for example SmartSnippet workspace folder). Open the terminal and navigate to DKF folder. The project contains make file that takes over the compilation process. The firmware will be compiled by invoking make command. Programming the DK board is done by invoking make command with firstflash parameter.

Figure 6.12: Programming Dev Kit board under Linux OS.
7. Product specification

The LPWAN Dev Kit is designed for enhanced LPWAN performance and manageability. In this chapter we briefly introduce the specifications for both hardware and software.

7.1 Software environment

To facilitate an easy network deployment, we have included many software features, which include but are not limited to:

- Open source SDK and software support
- Over The Air software update
- Mobile App for Android
- Free cloud service for managing and visualizing sensors data

7.2 Hardware environment

The MatchX Dev Kit is mainly designed for developers and designers for evaluation purposes. It helps to kickstart your project by providing test and evaluation hardware for proof of concept and enables programmers to develop software before custom hardware is ready. Using the exchangeable sensors provides incredible flexibility so the Dev Kit can be used for many different applications.
Chapter 7. Product specification

### Table 7.1: Key hardware specifications

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU</td>
<td>DA14680, 0 Hz up to 96 MHz 32-bit ARM Cortex-M0</td>
</tr>
<tr>
<td>Memory</td>
<td>8Mb Flash, 64kB OTP, 128kB ROM, 144kB SRAM</td>
</tr>
<tr>
<td>Interfaces</td>
<td>I²C, I²S, PCM, SPI, UART, USB, GPIOs</td>
</tr>
<tr>
<td>Wireless</td>
<td>Bluetooth 4.1 and LoRa</td>
</tr>
<tr>
<td>Battery</td>
<td>2.0mm pitch connector</td>
</tr>
<tr>
<td>Size</td>
<td>32 x 148 x 32mm (including Hat)</td>
</tr>
</tbody>
</table>

7.2.1 RF performance

There are two RF systems in the module, which include Lora, and Bluetooth. In this section we briefly introduce the performance of these systems. For Lora, both the "transmission" and "receive" performance are listed in Table 7.2 and "Bluetooth" can be found in Table 7.3.

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX Max</td>
<td>+18.5dBm</td>
</tr>
<tr>
<td>RX</td>
<td>down to -148dBm</td>
</tr>
</tbody>
</table>

Table 7.2: Lora RF performance

For Bluetooth is listed in Table 7.3.

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Power</td>
<td>0dBm</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>-94dBm</td>
</tr>
</tbody>
</table>

Table 7.3: Bluetooth performance.

7.2.2 Electrical characteristics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>V_BATT</td>
<td>Battery voltage</td>
<td>2.7</td>
<td>4.2</td>
<td>V</td>
</tr>
<tr>
<td>V_SNRP</td>
<td>Voltage output on the USB-C A8 and B8 pins</td>
<td>V_BATT</td>
<td>3.3</td>
<td>V</td>
</tr>
<tr>
<td>I_SNRP</td>
<td>Current output of 3V3_SNR</td>
<td>0</td>
<td>300mA</td>
<td>mA</td>
</tr>
<tr>
<td>V_GPIO</td>
<td>Voltage on any GPIO pin on USB-C</td>
<td>0</td>
<td>V_SNR</td>
<td>V</td>
</tr>
<tr>
<td>V_BUS</td>
<td>USB charging voltage</td>
<td>4.2</td>
<td>5.75</td>
<td>V</td>
</tr>
<tr>
<td>I_BUS</td>
<td>USB charging current supply</td>
<td>300</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td>T_op</td>
<td>Operating Temperature</td>
<td>-40</td>
<td>+85</td>
<td>°C</td>
</tr>
</tbody>
</table>

Table 7.4: Operating Range.
### 7.2 Hardware environment

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{\text{IDLE}}$</td>
<td>Current consumption, MCU awake, no RF activity</td>
<td>10</td>
<td>75</td>
<td>mA</td>
</tr>
<tr>
<td>$I_{\text{SEND}}$</td>
<td>Current consumption, sending LoRa packet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{\text{SLEEP}}$</td>
<td>Current consumption in sleep mode</td>
<td>&lt;10</td>
<td></td>
<td>µA</td>
</tr>
</tbody>
</table>

Table 7.5: Current consumption of the core module.

#### 7.2.3 Antenna characteristics

The SoM module is equipped with two U.FL connectors: 2.4GHz for Bluetooth and one for 868MHz (915MHz in US version) LoRa antenna. The parameters of the recommended antennas can be found in Table 7.6.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>2.4GHz antenna</th>
<th>868MHz (EU version)</th>
<th>915MHz (US version)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center Frequency</td>
<td>2.44GHz</td>
<td>868MHz</td>
<td>915MHz</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>101MHz</td>
<td>40MHz</td>
<td>40MHz</td>
</tr>
<tr>
<td>Gain</td>
<td>4.3dBi</td>
<td>2.33dBi</td>
<td>2.3dB</td>
</tr>
</tbody>
</table>

Table 7.6: Parameters of recommended antennas.
7.3 Dimensions

Figure 7.1: Dimension of the Dev Kit (top view), all dimensions in mm.
7.4 Certification

CE and FCC certification pending.