PiCAN 2 USER GUIDE

V1.3

Product name  PICAN CAN-Bus Board for Raspberry Pi
Model number  RSP-PICAN2
Manufacturer  SK Pang Electronics Ltd
Contents

Table of Contents

1. Introduction .................................................................................................................. 3
   1.1. Features ................................................................................................................. 3

2. Hardware Installation .................................................................................................... 3
   1.2. Configuring DB9 Connector .................................................................................. 4
   1.3. OBDII Cable .......................................................................................................... 4
   1.4. CAN Cable ............................................................................................................ 4
   1.5. Screw Terminal ..................................................................................................... 4
   1.6. 120Ω Terminator .................................................................................................. 5
   1.7. LED ....................................................................................................................... 5
   1.8. Not Fitted Items ..................................................................................................... 5

3. Software Installation ..................................................................................................... 6
   1.9. Bring Up the Interface .......................................................................................... 7

4. Writing Your Own Software ........................................................................................ 8
   1.10. Application in Python .......................................................................................... 8
   1.11. Application in C ................................................................................................... 9
1. Introduction
This PiCAN board provide CAN-Bus capability for the Raspberry Pi 2. It uses the Microchip MCP2515 CAN controller with MCP2551 CAN transceiver. Connections are made via DB9 or 4 way screw terminal. This board is also available with a 5v 1A SMPS that can power the Pi is well via the screw terminal or DB9 connector.

Easy to install SocketCAN driver. Programming can be done in C or Python.

1.1. Features
- CAN v2.0B at 1 Mb/s
- High speed SPI Interface (10 MHz)
- Standard and extended data and remote frames
- CAN connection via standard 9-way sub-D connector or screw terminal
- Compatible with OBDII cable
- Solder bridge to set different configuration for DB9 connector
- 120Ω terminator ready
- Serial LCD ready
- LED indicator
- Foot print for two mini push buttons
- Four fixing holes, comply with Pi Hat standard
- SocketCAN driver, appears as can0 to application
- Interrupt RX on GPIO25

2. Hardware Installation
Before installing the board make sure the Raspberry is switched off. Carefully align the 40way connector on top of the Pi. Use spacer and screw (optional items) to secure the board.
1.2. Configuring DB9 Connector
The CAN connection can be made via the DB9 connector. The connector be configured for different pinout. Depend if you are using an OBDII cable or a CAN cable.

1.3. OBDII Cable
Close the solder bridges on the lefthand side on SJ1, SJ2 and SJ3 as shown with a red dot.

1.4. CAN Cable
Close the solder bridges on the righthand side on SJ1, SJ2 and SJ3 as shown with a green dot.

<table>
<thead>
<tr>
<th>DB9 Pin number</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>GND</td>
</tr>
<tr>
<td>3</td>
<td>CAN_H</td>
</tr>
<tr>
<td>5</td>
<td>CAN_L</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DB9 Pin number</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>GND</td>
</tr>
<tr>
<td>7</td>
<td>CAN_H</td>
</tr>
<tr>
<td>2</td>
<td>CAN_L</td>
</tr>
</tbody>
</table>
1.5. Screw Terminal
The CAN connection can also be made via the 4 way screw terminal.

<table>
<thead>
<tr>
<th>Pin number</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CAN_H</td>
</tr>
<tr>
<td>2</td>
<td>CAN_L</td>
</tr>
<tr>
<td>3</td>
<td>GND</td>
</tr>
<tr>
<td>4</td>
<td>+12v In</td>
</tr>
</tbody>
</table>

Note: The +12v In is only used on the PiCAN2 board with SMPS option fitted.

1.6. 120Ω Terminator
There is a 120Ω fitted to the board. To use the terminator solder a 2way header pin to JP3 then insert a jumper.

1.7. LED
There is a red LED fitted to the board. This is connected to GPIO22.

1.8. Not Fitted Items
The board has footprint for two mini push buttons S1 and S2, they are connected to GPIO24 and GPIO23 respectively.

JP5 can be use to power a serial LCD with data on TXD line from the Pi. There is also 5v supply on JP5.

Switch mode power supply, this is a 5v module that can power the Pi. It has an input voltage range of 6v to 20v.

U2 is a EEPROM for ID use.
3. Software Installation

It is best to start with a brand new Raspbian image. Download the latest from:


After first time boot up, do an update and upgrade first.

```
sudo apt-get update
sudo apt-get upgrade
sudo reboot
```

Add the overlays by:

```
sudo nano /boot/config.txt
```

Add these 3 lines to the end of file:

```
dtparam=spi=on
 dtoverlay=mcp2515-can0,oscillator=16000000,interrupt=25
dtoverlay=spi-bcm2835-overlay
```

Reboot Pi:

```
sudo reboot
```
1.9. Bring Up the Interface

You can now bring the CAN interface up:

```bash
sudo /sbin/ip link set can0 up type can bitrate 500000
```

Download and copy the CAN test programs to the Pi.

[http://www.skpang.co.uk/dl/can-test_pi2.zip](http://www.skpang.co.uk/dl/can-test_pi2.zip)

Connect the PiCAN2 to your CAN network via screw terminal or DB9.

To send a CAN message use:

```
./cansend can0 7DF#02010500000000
```

This will send a CAN ID of 7DF. Data 02 01 05 – coolant temperature request.

Connect the PiCAN to a CAN-bus network and monitor traffic by using command:

```
./candump can0
```

You should see something like this:
4. Writing Your Own Software
You can write your own application software in either C or Python.

1.10. Application in Python
Download the Python-CAN files from:

https://bitbucket.org/hardbyte/python-can/get/4085cffe2519.zip

Unzip and install by

```
sudo python3 setup.py install
```

Bring the CAN interface up if it is not already done:

```
sudo /sbin/ip link set can0 up type can bitrate 500000
```

Now start python3

```
python3
```

To send a message out type the following lines:

```python
import can
bus = can.interface.Bus(channel='can0', bustype='socketcan_native')
msg = can.Message(arbitration_id=0x7de,
                   data=[0, 25, 0, 1, 3, 1, 4, 1],
                   extended_id=False)
bus.send(msg)
```
To received messages and display on screen type:

```python
notifier = can.Notifier(bus, [can.Printer()])
```

1.11. Application in C

Bring the CAN interface up if it is not already done:

```bash
sudo /sbin/ip link set can0 up type can bitrate 500000
```

Download the source code and example files by typing the following in the command prompt:

```bash
wget http://skpang.co.uk/dl/cantest.tar
```

Unpack the tar file and change into directory by:

```bash
tar xf cantest.tar
```

```bash
cd linux-can-utils
```

The example file is called cantest.c to edit this file, type the following in the command prompt:

```bash
nano cantest.c
```

Line 77 is the CAN message to be sent out.

```c
unsigned char buff[] = "7DF#0201050000000000";
```

7DF is the message ID and 0201050000000000 is the data. Change the data to suit.

Press CTRL-X to exit.

To compile the program type:

```bash
make
```

Check there are no errors. To run the program type:

```bash
./cantest
```