Supporting High Power Mode on a JN5189/QN9090/K32W device.

1 Application Note Overview
The following devices
   • JN5189(T)/JN5188(T)
   • QN9090(T)/QN9030(T)
   • K32W061/K32W041
feature an integrated radio, which can be used with an external Front End. This application note is concerned with setting up the device for hardware designs that uses a FEM (Front End module).

1.1 Contents

Contents
1 Application Note Overview ................................................................................................................................. 1
1.1 Contents ....................................................................................................................................................... 1
2 FEM Design.................................................................................................................................................. 2
3 What is Required ......................................................................................................................................... 2
3.1 Setting up the DIO .................................................................................................................................. 2
4 Zigbee 3.0 Applications ............................................................................................................................... 3
4.1 Setting up the Radio Parameters ............................................................................................................. 3
4.2 Editing the Zigbee 3.0 Application Notes ............................................................................................... 4
5 BLE Applications ........................................................................................................................................ 4
5.1 Editing the BLE Example Applications ................................................................................................. 4
Revision History .............................................................................................................................................. 5
2 FEM Design

The parameters used in this example are suitable for a test design based on a Skyworks SKY66112-11. The exact parameters needed for compliance with different Front Ends or different layouts will require additional testing to confirm that the parameters used are compliant. The examples design used is shown below:

3 What is Required

To support a Front-End module the following is required

1) Set up RX and TX DIO control
2) Set the compliance limit parameter on the radio

3.1 Setting up the DIO

Some of the PIO lines can be used to generate RFRX and RFTX signals. These signals are connected directly from the MODEM with the device and indicate if the device is in RX or TX mode and, by implication, if the MODEM is idle. Any of the PIOs that have RFRX and RFTX options can be used from the table below:

<table>
<thead>
<tr>
<th>I/O Names</th>
<th>FUNC0</th>
<th>FUNC1</th>
<th>FUNC2</th>
<th>FUNC3</th>
<th>FUNC4</th>
<th>FUNC5</th>
<th>FUNC6</th>
<th>FUNC7</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIO0_0</td>
<td>GPIO0</td>
<td>USART0_SCK</td>
<td>USART1_TXD</td>
<td>-</td>
<td>PWM5-PU</td>
<td>SPI1_SCK</td>
<td>-</td>
<td>PDM0_DATA</td>
</tr>
<tr>
<td>PIO0_1</td>
<td>GPIO1</td>
<td>USART1_RXD</td>
<td>USART0_TXD</td>
<td>-</td>
<td>PWM1-PD</td>
<td>SPI1_MISO</td>
<td>-</td>
<td>PDM0_CLK</td>
</tr>
<tr>
<td>PIO0_2</td>
<td>GPIO2</td>
<td>SPI0_SCK</td>
<td>USART0_SCK</td>
<td>-</td>
<td>PWM2-PO</td>
<td>SPI1_MOSI</td>
<td>ISO7816_RST</td>
<td>MCLK</td>
</tr>
<tr>
<td>PIO0_3</td>
<td>GPIO3</td>
<td>SPI0_MISO</td>
<td>USART0_CTS</td>
<td>-</td>
<td>PWM3-PU</td>
<td>SPI1_SEELN0</td>
<td>ISO7816_CLK</td>
<td>-</td>
</tr>
<tr>
<td>PIO0_4</td>
<td>GPIO4</td>
<td>SPI0_MOSI</td>
<td>USART0_TXD</td>
<td>-</td>
<td>PWM4-PU</td>
<td>SPI1_SEELN1</td>
<td>ISO7816_ID</td>
<td>RFTX</td>
</tr>
<tr>
<td>PIO0_5</td>
<td>GPIO5</td>
<td>SPI0_MOSI</td>
<td>USART0_RTS</td>
<td>-</td>
<td>SPI0_MISO</td>
<td>SPI1_SEELN2</td>
<td>-</td>
<td>RFRX</td>
</tr>
<tr>
<td>PIO0_6</td>
<td>GPIO6</td>
<td>SPI0_SSELN</td>
<td>USART0_RTS</td>
<td>CT32B1_MAT0</td>
<td>PWM5-PO</td>
<td>2C1_SCL</td>
<td>USART1_TXD</td>
<td>ADE</td>
</tr>
<tr>
<td>PIO0_7</td>
<td>GPIO7</td>
<td>SPI0_MISO</td>
<td>USART0_CTS</td>
<td>CT32B1_MAT1</td>
<td>PWM7-PO</td>
<td>2C1_SDA</td>
<td>USART1_RXD</td>
<td>ADO</td>
</tr>
<tr>
<td>PIO0_8</td>
<td>GPIO8</td>
<td>SPI0_MOSI</td>
<td>USART0_TXD</td>
<td>CT32B0_MAT0</td>
<td>PWM8-PO</td>
<td>2C1_SDA</td>
<td>USART1_TXD</td>
<td>ADE</td>
</tr>
<tr>
<td>PIO0_9</td>
<td>GPIO9</td>
<td>SPI0_SSELN</td>
<td>USART0_RXD</td>
<td>CT32B1_CAP1</td>
<td>PWM9-PO</td>
<td>2C1_SCL</td>
<td>USART1_TXD</td>
<td>ADO</td>
</tr>
<tr>
<td>PIO10</td>
<td>GPIO10</td>
<td>CT32B0_CAP0</td>
<td>USART1_TXD</td>
<td>-</td>
<td>RFTX</td>
<td>2C2_SCL</td>
<td>SPI0_SCK</td>
<td>PDM0_DATA</td>
</tr>
<tr>
<td>PIO11</td>
<td>GPIO11</td>
<td>CT32B1_CAP0</td>
<td>USART1_RXD</td>
<td>-</td>
<td>RFTX</td>
<td>2C2_SDA</td>
<td>SPI0_MISO</td>
<td>PDM0_CLK</td>
</tr>
<tr>
<td>PIO12</td>
<td>GPIO12</td>
<td>IR_BLASTER</td>
<td>SWCLK</td>
<td>-</td>
<td>PDM0-PD</td>
<td>2C1_SCL</td>
<td>PDM0_MISO</td>
<td>ACOMP_OUT</td>
</tr>
<tr>
<td>PIO13</td>
<td>GPIO13</td>
<td>SPI0_SSELN2</td>
<td>SWDIO</td>
<td>-</td>
<td>PDM0-PD</td>
<td>2C1_SDA</td>
<td>SPI0_SSELN</td>
<td>-</td>
</tr>
<tr>
<td>PIO14</td>
<td>GPIO14</td>
<td>SPI0_SSELN1</td>
<td>USART0_SCK</td>
<td>CT32B0_CAP1</td>
<td>PWM5-PU</td>
<td>2C0_SCL</td>
<td>MCLK</td>
<td>RFTX</td>
</tr>
<tr>
<td>PIO15</td>
<td>GPIO15</td>
<td>SPI0_SSELN1</td>
<td>ANA_COMP_OUT</td>
<td>-</td>
<td>PWM3-PU</td>
<td>2C0_SDA</td>
<td>PDM1_DATA</td>
<td>RFRX</td>
</tr>
<tr>
<td>PIO16</td>
<td>GPIO16</td>
<td>SPI0_SSELN0</td>
<td>ISO7816_RST</td>
<td>-</td>
<td>PWM5-PU</td>
<td>2C0_SDA</td>
<td>PDM1_CLK</td>
<td>SPI1_CSN</td>
</tr>
<tr>
<td>PIO17</td>
<td>GPIO17</td>
<td>SPI0_MISO</td>
<td>ISO7816_CLK</td>
<td>-</td>
<td>PWM5-PO</td>
<td>CLK_OUT</td>
<td>-</td>
<td>SPI1_E03</td>
</tr>
<tr>
<td>PIO18</td>
<td>GPIO18</td>
<td>SPI0_MISO</td>
<td>ISO7816_ID</td>
<td>CT32B0_MAT1</td>
<td>PWM7-PO</td>
<td>USART0_TXD</td>
<td>-</td>
<td>SPI1_CLK</td>
</tr>
<tr>
<td>PIO19</td>
<td>GPIO19</td>
<td>ADD</td>
<td>USART1_RXD</td>
<td>CLK_IN</td>
<td>PWM4-PO</td>
<td>USART0_RXD</td>
<td>-</td>
<td>SPI1_E00</td>
</tr>
<tr>
<td>PIO20</td>
<td>GPIO20</td>
<td>IR_BLASTER</td>
<td>USART1_TXD</td>
<td>-</td>
<td>PWM8-PO</td>
<td>RFTX</td>
<td>-</td>
<td>SPI1_E02</td>
</tr>
<tr>
<td>PIO21</td>
<td>GPIO21</td>
<td>IR_BLASTER</td>
<td>USART1_SCK</td>
<td>-</td>
<td>PWM8-PU</td>
<td>RFRX</td>
<td>-</td>
<td>SPI1_E01</td>
</tr>
</tbody>
</table>
Special PIO functions are setup using IOCON_PinMuxSet. To use this function, include the following header

```
#include "fsl_iocon.h"
```

In our example, the required functions are on DIO4 and DIO5 using IOCON_FUNCTION 7. This is done with the following two lines of code:

```
IOCON_PinMuxSet(IOCON, 0, 4, IOCON_MODE_INACT | IOCON_FUNC7 | IOCON_DIGITAL_EN);
IOCON.PinMuxSet(IOCON, 0, 5, IOCON_MODE_INACT | IOCON_FUNC7 | IOCON_DIGITAL_EN);
```

## 4 Zigbee 3.0 Applications

### 4.1 Setting up the Radio Parameters

There are two functions that are defined to set up the radio. Both should be used in conjunction to configure the radio. These functions are defined in AppApi.h.

The first function is

```
PUBLIC void vAppApiSetRadioTxModes(teRadioTxMode eTxMode, teRadioTxMode eTxModeCh26);
```

This is concerned with changing the radios internal settings to set the radio is in different proprietary modes affect currents and setting within the radio. The following are defined

```
E_RADIO_TX_MODE_STD    = 0
E_RADIO_TX_MODE_PROP_1 = 1
E_RADIO_TX_MODE_PROP_2 = 2
```

The standard way to use this would be to set the radio XT mode to standard. If there are issues on radio channel 26 with band edge due to the output power from the front end then using proprietary mode for channel 26 may be required. In our example, the following is used:

```
vAppApiSetRadioTxModes(E_RADIO_TX_MODE_STD, E_RADIO_TX_MODE_PROP_2);
```

The second function is

```
PUBLIC void vAppApiSetComplianceLimits(int8  i8TxMaxPower, int8  i8TxMaxPowerCh26, uint8 u8CcaThreshold);
```

This function deals with standard parameters for the radio such as CCA threshold. These parameters should be evaluated using CMET.

In our example, the radio is set to output 4db and the CCA threshold is adjusted to 83 to deal the RX gain in the LNA and still be compliant with the IEEE802.15.4 specification. The following is used:

```
vAppApiSetComplianceLimits(4, 4, 83);
```
4.2 Editing the Zigbee 3.0 Application Notes

The code need to be added before BEFORE ZPS_eAplAfInit() is called. In ‘JN-AN-1244-ZigBee-3.0-Light-Bulbs-for-JN518x’, the most suitable location is in APP_vInitialiseNode() in app_zlo_light_node.c. Add the following code before ZPS_eAplAfInit() is called:

```c
// Start of HP Support Initialisation
IOCON_PinMuxSet(IOCON, 0, 4, IOCON_MODE_INACT | IOCON_FUNC7 | IOCON_DIGITAL_EN);
IOCON.PinMuxSet(IOCON, 0, 5, IOCON_MODE_INACT | IOCON_FUNC7 | IOCON_DIGITAL_EN);
vAppApiSetComplianceLimits(4, 4, 83);
vAppApiSetRadioTxModes(E_RADIO_TX_MODE_STD, E_RADIO_TX_MODE_PROP_2);
// End of HP Support Initialisation
...
// Existing Code
ZPS_u32MacSetTxBuffers (4);
/* Initialise ZBPro stack */
ZPS_eAplAfInit();
```

5 BLE Applications

5.1 Editing the BLE Example Applications

The Code need to be added within the BleApp_Init function. For example, in ‘beacon_bm’, the code should be added at the end of void BleApp_Init(void) in beacon.c:

```c
// Start of HP Support Initialisation
IOCON_PinMuxSet(IOCON, 0, 4, IOCON_MODE_INACT | IOCON_FUNC7 | IOCON_DIGITAL_EN);
IOCON_PinMuxSet(IOCON, 0, 5, IOCON_MODE_INACT | IOCON_FUNC7 | IOCON_DIGITAL_EN);
```
Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1V0</td>
<td>First release</td>
</tr>
<tr>
<td>1V1</td>
<td>Added BLE support</td>
</tr>
<tr>
<td>1V2</td>
<td>Added K32W</td>
</tr>
</tbody>
</table>

How To Reach Us

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

Home Page:

nxp.com

Web Support:

nxp.com/support

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetics, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QoriQ, QoriQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQi, Vybrid, Afast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V.

All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© NXP B.V. 2020. All rights reserved.
For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 04/2020
Document identifier: JN-AN-1252