



WyzBee™

**User Manual** 

Version 1.2

October 2015

Redpine Signals, Inc.

2107 N. First Street, #680 San Jose, CA95131. Tel: (408) 748-3385 Fax: (408) 705-2019 Email: <u>info@redpinesignals.com</u> Website: <u>www.redpinesignals.com</u>





#### About this Document

This document describes general information of WyzBee<sup>™</sup> along with the board bring up and installation procedure for software tools for developing applications, including a sample TriLED demo programmed using Keil IDE platform. This document elaborates all the features and steps for using the WyzBee<sup>™</sup> platform.

#### **Disclaimer:**

The information in this document pertains to information related to Redpine Signals, Inc. products. This information is provided as a service to our customers, and may be used for information purposes only. Redpine assumes no liabilities or responsibilities for errors or omissions in this document. This document may be changed at any time at Redpine's sole discretion without any prior notice to anyone. Redpine is not committed to updating this document in the future.

Copyright © 2015 Redpine Signals, Inc. All rights reserved.





## Table of Contents

1	Intr	oduction5
	1.1	Features5
	1.1.1	1 MCU Features5
	1.1.2	2 WLAN Features
	1.1.3	3 Bluetooth6
	1.1.4	4 ZigBee6
	1.1.5	5 General6
	1.2	WyzBee <sup>™</sup> Board7
	1.3	WyzBee™ Top View8
	1.4	WyzBee™ Bottom View8
2	Set	ting up WyzBee™9
	2.1	Setup Requirements
	2.2	Installing the CMSIS-DAP Drivers9
	2.3	Installing IDEs
3	Get	ting Started11
	3.1	Starting a New Project
	3.2	Configuring IDE for WyzBee™
	3.3	Sample Project
	3.4	Adding THINGS
Δ	Δητ	pendix Δ· Installing IDFs 18
-	7 PF 1 1	Koil IDE
	<b>4.1</b>	1 Download and Installation 18
	412	2 Configuring Keil IDE for WyzBee™ 21
	4.2	ColDE 21
	4.2.3	22 Download and Installation
	4.2.2	2 Configuring CoIDE for WyzBee™24
	4.3	IAR Embedded Workbench
	4.3.2	1 Download and Installation27
	4.3.2	2 Configuration IAR Embedded Workbench for WyzBee™35
5	Арр	oendix B: WyzBee™ Headers37
	5.1	Pin Description

## Table of Figures

Figure 1: WyzBee™ Baseboard with micro-B USB Cable	7
Figure 2: WyzBee™ Baseboard's Top View	8
Figure 3: WyzBee™ Baseboard's Bottom View	8
Figure 4: CMSID-DAP Drivers Installation Window	9
Figure 5: WyzBee™ CMSIS-DAP Debug Port Detection	10
Figure 6: Keil uVision: Selecting the Device	11
Figure 7: Keil uVision: Options for Target – Utilities	12
Figure 8: Keil uVision: CMSIS-DAP Debugger Settings	12
Figure 9: Keil uVision: Options for Target – Debug	13
Figure 10: Keil uVision: Options for Target – Debug	13
Figure 11: Keil uVision: Options for Target – Target	14
Figure 12: Keil uVision: Options for Target – Output	14



Figure 13: Keil uVision: Options for Target – Linker	15
Figure 14: main.c of TriLED Project in Keil uVision4	16
Figure 15: TriLED_App Function	16
Figure 16: Debug Cursor at main Function	17
Figure 17: Keil IDE Installation Window	18
Figure 18: Keil IDE Installation License Agreement	19
Figure 19: Keil IDE Installation Folder Selection	19
Figure 20: Keil IDE Customer Information	20
Figure 21: Keil IDE Installation Completed	20
Figure 22: Keil IDE Installation Completed	21
Figure 23: ULINK Driver Installation	21
Figure 24: CoIDE Installation Window	22
Figure 25: CoIDE Installation Folder Selection	23
Figure 26: CoIDE Installation Start	23
Figure 27: GCC Toolchain Path for CoIDE	24
Figure 28: CoIDE Version	24
Figure 28: CoIDE Configuration Button	24
Figure 29: CoIDE Configuration – Device	25
Figure 30: CoIDE Configuration – Link	25
Figure 31: CoIDE Configuration – Output	26
Figure 32: CoIDE Configuration – User	26
Figure 33: CoIDE Configuration – Debugger	26
Figure 34: CoIDE Configuration – Download	27
Figure 35: IAR Embedded Workbench Installation Window	28
Figure 36: IAR Embedded Workbench Installation Window	28
Figure 37: IAR Embedded Workbench License Agreement	29
Figure 38: IAR Embedded Workbench Installation Folder	29
Figure 39: IAR Embedded Workbench Installation Start	30
Figure 40: IAR Embedded Workbench – Dongle Drivers Installation	30
Figure 41: IAR Embedded Workbench Installation Completed	31
Figure 42: IAR Embedded Workbench – Device Software – 1	31
Figure 43: IAR Embedded Workbench – Device Software – 2	32
Figure 44: IAR Embedded Workbench – Language Selection	32
Figure 45: IAR Embedded Workbench – License Wizard	32
Figure 46: IAR Embedded Workbench – License Wizard (Choose a Product)	33
Figure 47: IAR Embedded Workbench – License Wizard (Register)	33
Figure 48: IAR Embedded Workbench: Registration	34
Figure 49: IAR Embedded Workbench: Registration Complete	35
Figure 50: IAR Embedded Workbench – Project Options	35
Figure 51: IAR Embedded Workbench – Options – 1	36
Figure 52: IAR Embedded Workbench – Options – 2	36
Figure 53: 32-pin WyzBee™ THING Header	37

## Table of Tables

ble 1: TriLED Connections to MCU15
------------------------------------



## 1 Introduction

The WyzBee<sup>™</sup>IoT platform is an industry-first single source offering with fully inclusive sensing, computing, communicating, and cloud support. The compact WyzBee<sup>™</sup> board includes Redpine's Wireless Secure MCU (WiSeMCU<sup>™</sup>) with multi-protocol wireless module providing Wi-Fi, Bluetooth 4.1, and ZigBee connectivity, nine-axis inertial sensors, onboard temperature/humidity sensors, 3 axis accelerometer , an infrared receiver, a USB (debug) port, push-buttons, LEDs, and WyzBee<sup>™</sup> THING<sup>™</sup> expansion connector. The WiSeMCU module runs an embedded TCP/IP networking stack with SSL/TLS/HTTPS security, apart from complete Wi-Fi, BT 4.1, and ZigBee stacks.

The WyzBee<sup>™</sup> THING expansion headers accommodate a host of other symbiotic devices, with a number of peripherals – called 'THINGS' – already available from Redpine including audio, GSM, GPS, capacitive touch display, rechargeable battery, and additional sensors. Application development is supported with a choice of development environments – IAR, Keil, and the free CoIDE from CooCox.

The WiSeMCU<sup>™</sup> module integrates PUF-based hardware security block that provides for unique, individual device entities – ensuring that each IoT device can be individually authenticated and software delivered to it that cannot run on any other device.

## 1.1 Features

## 1.1.1 MCU Features

- ARM Cortex-M4F processor, running at a frequency of up to 160 MHz
- Integrated Floating Point Unit (FPU), Memory Protection Unit (MPU), Built-in Nested Vectored Interrupt Controller (NVIC)
- Debug options: JTAG and Embedded Trace Macrocells (ETM)
- 1MB on-chip flash program memory with flash accelerator and 32KB work flash memory
- 128 KB SRAM for code and data use
- CAN Interface with support for up to 2 channels
- Up to 32 high speed general purpose I/O ports.
- Multi-function Serial Interface with support for up to 6 channels (UART, CSIO (SPI), and I2C).
- Base timer (maximum 8 channels) supporting PWM, PPG, reload timer, PWC (up to 6 channels).
- Comprehensive Timers: Multi-function timer (MFT) with FRT, WFG, ICU, OCU modes supported.
- RTC, QPRC, Dual Timer supported.
- Up to 2 configurable Watchdog Timers.
- Analog peripherals: 12-bit, 11-channel Analog-to-Digital Converter (ADC)
- Security: Unique ID of the device (41 bit) is set
- Six low-power consumption modes: SLEEP, Timer, RTC, STOP, Deep Standby RTC, Deep Standby stop.





- 12-bit Digital to Analog Converter (DAC) with support for 1 channel
- DMA Controller with support for up to 8 channels
- CRC (Cyclic Redundancy Check) Accelerator
- External interrupt input pins: up to 14 pins
- Low-Voltage Detector (LVD) current: 100nA
- CMSIS-DAP Debug adapter: WyzBee<sup>™</sup> comes with an on board CMSIS-DAP debug adapter for downloading and debugging applications, without the need for an external debugger.

#### 1.1.2 WLAN Features

- Compliant to single-spatial steam IEEE 802.11 a/b/g/n with dual band (2.4 and 5 GHz) support.
- Support for 20MHz channel bandwidth.
- Transmit power up to +18dBm with integrated PA.
- Receive sensitivity of -97dBm.
- Supports Wi-Fi Direct<sup>™</sup>, Access point mode, WPA/WPA2-PSK, WPA/WPA2-Enterprise (EAP-TLS, EAP-FAST, EAP-TTLS, PEAP-MS-CHAP-V2).

#### 1.1.3 Bluetooth

- Compliant to dual-mode Bluetooth 4.0.
- Transmit power up to 15dBm (class-1) with integrated PA.
- Receive sensitivity of -94 dBm.
- Basic Bluetooth profile embedded in device.

#### 1.1.4 ZigBee

- Compliant to IEEE 802.15.4
- Transmit power up to 15 dBm with integrated PA.
- Receive sensitivity of -102 dBm.
- ZigBee Pro stack embedded

#### 1.1.5 General

- U.FL connector for external antenna connection.
- Operating temperature range:  $-40^{\circ}$ C to  $+85^{\circ}$ C
- TCP/IP stack (IPv4/IPv6), HTTP/HTTPS, DHCP, ICMP, SSL 3.0/TLS1.2, Web sockets, IGMP, FTP Client, SNTP, DNS, embedded in the device.
- On Board Peripherals:
  - o Tri Color LED

LED Color	Pin
Red	P41





Blue	P3F
Green	P3E

• Two push buttons (One for Reset and one for an External interrupt)

Push Button	Pin
SW1(External Interrupt)	P50
SW2 (Reset)	MCU_RESET_N

o IR receiver

IR Receiver	Pin
IR_OUT	P42

- o 9 Axis Sensor
- o 3 Axis Accelerometer
- o Humidity and Temperature Sensor

Sensor	Pin		
All three sensors use the I2C interface to interact with the MCU.			
Clock	Р33		
Data	Р32		

• Micro-B USB Full Speed Interface

## **1.2** WyzBee<sup>™</sup> Board

WyzBee<sup>™</sup> is a USB-powered device. Shown below is a WyzBee<sup>™</sup> baseboard with the micro-B USB cable plugged in.



Figure 1: WyzBee<sup>™</sup> Baseboard with micro-B USB Cable

Page 7





## 1.3 WyzBee<sup>™</sup> Top View



Figure 2: WyzBee<sup>™</sup> Baseboard's Top View

## 1.4 WyzBee<sup>™</sup> Bottom View



Figure <u>3:</u> WyzBee<sup>™</sup> Baseboard's Bottom View



## 2 Setting up WyzBee<sup>™</sup>

## 2.1 Setup Requirements

Setting up and working with WyzBee<sup>™</sup> is easy. Before you start, make sure you have the following hardware and software components:

Hardware:

- The WyzBee<sup>™</sup> baseboard
- 32/64-bit PC with minimum 2GB RAM and USB Port for Power, Downloading and Debugging

Software:

- Windows 7/8/8.1 Operating System.
- The CMSIS-DAP driver provided by Spansion. You can download the driver from <u>https://www.spansion.com/Support/microcontrollers/developmentenvironment/</u> <u>Pages/board-SK-FM4-U120-9B560.aspx</u>. More details are given in <u>Section 2.2</u>.
- An IDE like Keil uVision, IAR Embedded Workbench or CooCox CoIDE.

## 2.2 Installing the CMSIS-DAP Drivers

Follow the steps below to download and install the CMSIS-DAP Drivers.

1) Download the complete set of drivers and tools from the following link:

(USB drivers for virtual COM port and CMSIS-DAP)

https://www.spansion.com/Support/microcontrollers/developmentenvironment/Page s/board-SK-FM4-U120-9B560.aspx

**Note:** If the device gets detected as an unknown device, update the drivers from the device manager and point to the location of the newly downloaded drivers.

2) Windows USB drivers that are specific to WyzBee<sup>™</sup> need to be installed for the detection of debugging port. In the downloaded folder, navigate to the "drivers" folder and double-click on the driverinstaller.exe file. The window below will appear.



Figure 4: CMSID-DAP Drivers Installation Window



- 3) Click the Next button.
- 4) A "Publisher cannot be verified", warning might appear depending on the Windows Security Settings. Select the "Install this driver software anyway" option. This installs the cmsis-dap and usbdirect drivers.
- 5) Click the Finish button after the drivers' installation is completed.
- 6) Connect the micro-B USB cable between the WyzBee™ USB port and the PC. Verify that the board is detected under the "Ports" section of the Windows Device Manager.

	PCL Simple Communications Controller
	h Universal Serial Bus (USB) Controller
🍙 🖓 F	Ports (COM & LPT)
	CMSIS-DAP Spansion Virtual Communications Port (COM17)
	Communications Port (COMI)
	SUNIX LPT Port (LPT3)
Þ 🔲 F	Processors
þ 🚛 S	lystem devices

#### Figure 5: WyzBee<sup>™</sup> CMSIS-DAP Debug Port Detection

## 2.3 Installing IDEs

WyzBee<sup>™</sup> applications can be developed using various IDEs like CoIDE, Keil and IAR. Refer to <u>Appendix A</u> for instructions on downloading and installing these tools..



## 3 Getting Started

This section helps in getting started quickly with WyzBee<sup>™</sup>. It describes the process for starting a project, configuring the IDE for WyzBee<sup>™</sup> and then working with an example project.

It is necessary to know your way around one of the supported IDE's in order to work on WyzBee<sup>™</sup>. The steps below use the Keil uVision 4 IDE as an example to download and debug a sample application on WyzBee<sup>™</sup>.

## 3.1 Starting a New Project

- 1) Start the Keil uVision 4 IDE.
- 2) Click on "New uVision Project" from the "Project" dropdown menu. A popup window appears giving you an option to select the location of the project. Select the path you would like and click "Save".
- A new window opens for selecting the device. Scroll down the list and select "MB9BF568N" as shown below.

Vendor: Spansion Device: MB9BF568N Toolset: ARM		
Search:	Description:	
MB9BF529T     MB9BF566M     MB9BF566N     MB9BF566R     MB9BF567M     MB9BF567N     MB9BF567N     MB9BF568M     MB9BF568N     MB9BF568R     MB9BF568R     MB9BF568R     MB9BF616S		E F

## Figure 6: Keil uVision: Selecting the Device

4) Click OK and you are now ready to start configuring the IDE for WyzBee<sup>™</sup>.

## 3.2 Configuring IDE for WyzBee™

The Keil IDE is used here as an example. The process for CoIDE and IAR Embedded Workbench is explained in <u>Appendix A</u>.





- 1) In the Keil uVision IDE window, click on "Options for Target".
- 2) In the new window that opens, click on the "Utilities" tab. Uncheck the "Use Debug Driver" option.
- 3) Next, in the dropdown menu for "Use Target Driver for Flash Programming", select "CMSIS-DAP Debugger" and click "Settings".

Configure Flash M	enu Command		
Use Target Dr	river for Flash Programming		🧮 Use Debug Driver
C	ISIS-DAP Debugger	✓ Settings	✓ Update Target before Debugging
Init File:			Edit
Configure Image F	Run Independent ile Processing (FCARM):		
Uutput File:		Add Output	File to Group:
		Jeonmon	Generate Listing
Image Files Root	Folder:		

Figure 7: Keil uVision: Options for Target – Utilities

4) In the popup window, click "Add" and select the device name highlighted in the image below. Click OK.

C Erase Full Chip Erase Sectors C Do not Erase	<ul> <li>✓ Program</li> <li>✓ Verify</li> <li>✓ Reset and Ru</li> </ul>	n	20040000 Size: 0x0800	
Description	Device Size	Device Type	Address Range	
NESESSE TUZAKE Hash	TM	on-chip Hash	UUUUUUUUH - UUUPPPPPH	
		Start:	Size:	

Figure 8: Keil uVision: CMSIS-DAP Debugger Settings





5) Next, click the "Debug" tab and select "Use" and "CMSIS-DAP Debugger" as shown below.

C Use Simulat	or Sett	ngs
Limit Speed	to Real-Time	
<ul> <li>Load Applic nitialization File</li> </ul>	ation at Startup 🔽 Run to main()	<ul> <li>Load Application at Startup</li> <li>Initialization File:</li> </ul>
	Ed	t Edit
Restore Debu	g Session Settings	Restore Debug Session Settings
🔽 Breakpo	ints 🔽 Toolbox	🔽 Breakpoints 🔽 Toolbox
🔽 Watch V	Vindows & Performance Analyzer	Watch Windows
Memory	Display 🔽 System Viewer	Memory Display 🔽 System Viewer
CPU DLL:	Parameter:	Driver DLL: Parameter:
SARMCM3.DL	L -MPU	SARMCM3.DLL -MPU
Dialog DLL:	Parameter:	Dialog DLL: Parameter:
DCM.DLL	-pCM4	TCM.DLL pCM4

Figure 9: Keil uVision: Options for Target – Debug

- 🛛 Options for Target 'Target 1' × Device Target Output Listing User C/C++ Asm Linker Debug Utilities Generic CPU Data Base -Vendor: Spansion Device: MB9BF568N Toolset: ARM Search: 32bit ARM Cortex-M4F Core (r0p1) MB9BF568N . - Maximum clock frequency 160MHz . MB9BF568R - FPU built-in - Support DSP instruction MB9BF616S Memory: MB9BF616T -up to 1024kB Main Flash - 32kB Work Flash -up to 128kB RAM MB9BF617S MB9BF617T Base Timer: 8 channels(Max.) Watch counter CRC accelerator MB9BF618S MB9BF618T Multi-function Timer: 2 unit(Max.) - 16-bit free-run timer x3 channels/unit MB9BFD16S Input capture x4 channels/unit MB9BFD16T • Defaults OK Cancel Help
- 6) Click on the "Device" tab and ensure that the settings are as shown in the image below.

- Figure 10: Keil uVision: Options for Target Debug
- 7) Click on the "Target" tab and ensure all settings are as shown in the image below.





🕎 Options for Target 'TriLED_blin	k'					23
Device Target Output Listing	User C/C++ As	m Linker	Debug Utili	ties		
Spansion MB9BF568N						
	Xtal (MHz): 4.0		Generation —			
Operating system: None		-	Use Cross-Mod	dule Optimizat	tion	
System-Viewer File (.Sfr):	System-Viewer File (.Sfr): Use MicroLIB Big Endian					
MB9BF56xx.SFR		Floa	ting Point Har	dware:	Jse FPU	•
Use Custom SVD File						
Read/Only Memory Areas		Read	/Write Memory	y Areas ——		
default off-chip Start	Size Star	tup defaul	t off-chip	Start	Size	Nolnit
ROM1:	(		RAM1:			
ROM2:			RAM2:			
ROM3:	(		RAM3:			
on-chip	,		on-chip			
IROM1: 0x0	0x100000 @	<b>N</b>	IRAM1:	20038000	0x10000	
□ IROM2: 0x200C0000	0x8000		IRAM2: 0x	(1FFF0000	0x10000	
	OK	Cancel	Defaults	.		Help
		Carloci	Derduita	,	_	Holp

Figure 11: Keil uVision: Options for Target – Target

8) Click on the "Output" tab and ensure all settings are as shown in the image below.

🛛 Options for Target 'Target 1'		X
Device   Target Output   Listing   User   C/C++   Asm   Linker   Debug   Utilities		
Select Folder for Objects Name of Executable: http_app		
<ul> <li>Create Executable: .\http_app</li> <li>✓ Debug Information</li> </ul>	Create Batch File	
Create HEX Hie     F     Browse Information     Create Library: Vitto, ann lib		
Create Library: .vnttp_app.lib		
OK Cancel Defaults	He	lp

Figure 12: Keil uVision: Options for Target – Output

9) Click on the "Linker" tab and ensure all settings are as shown in the image below.





evice   Targe	et Output Listing User C/C++ Asm	Linker Debug	Utilities	
└ Use Mem	nory Layout from Target Dialog	X/O Base:		_
∏ Make	RW Sections Position Independent	R/O Base:	0x0000000	
Make	RO Sections Position Independent	R/W Base	0x2003C000	_
Don't	Search Standard Libraries	disable Warnings:		
Scatter File	http_app.sct		, 	Edit
Scatter File	http_app.sct			Edit
Scatter File Misc controls	http_app.sct			Edit

#### Figure 13: Keil uVision: Options for Target – Linker

The IDE is now configured for WyzBee<sup>™</sup>. You can now start writing your application!

#### 3.3 Sample Project

A sample project, TriLED.zip, which blinks the onboard TriLED is provided to help you quickly get started on WyzBee<sup>™</sup>. The details of the pins of the MCU connected to the TriLED are given below:

LED Color	MCU Pin
Red	P41
Green	РЗЕ
Blue	P3F

Table 1: TriLED Connections to MCU

The project, when run, blinks the LEDs one at a time with a certain delay. Follow the steps below to compile, flash and run the example project.

- 1) Download the blinky project (TriLED.zip) from the URL above and extract it.
- 2) Double-click the Keil uVision4 project file. This opens the Keil IDE.

\$(Extracted\_Folder)\TriLed\IDE\Keil\TriLed.uvproj

3) Open main.c in the IDE from the Project menu, as shown in the image below.





D:\Revision_8\projects\testing\keil\TriLed\IDE\Keil\TriLed.uvproj - µVision4				
File Edit View Project Flash	n Debug Peripherals Tools SVCS Window Help			
i 🗋 📴 🖬 🗿 🐰 🔈 🕵	(マ) (マーマ) (P) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2			
	▲ main.c			
E- C Applications	16 - 1/*			
Applications	18 *			
E Libraries	19 * main file			
🗄 🖃 main.c	21 * file : main.c 22 * Version : V1.00			
	23 * History :			
	24 * 2015-09-01 : V1.00 First version.			
	25 ************************************			
	27 ************************************			
	28 */			
	29			
	30			
	32   ' **********************************			
	33 * INCLUDE FILES			
	34 ***			
	36 -			
	37 #include <board_init.h></board_init.h>			
	38 #include <triled_app.h></triled_app.h>			
	39 40 int main()			
	42			
	43 Board_init();			
	45 TriLed App();			
	46			
	47 return 0;			
	48 40 1			
	50			

Figure 14: main.c of TriLED Project in Keil uVision4

4) The TriLED\_App() function controls all the frontend functioning of the application. It initializes the necessary drivers and configures WyzBee<sup>™</sup>. The source code is shown in the image below.

100 void TriLed_App(void)	
101 🖂 {	
102 while(1)	
103 🛱 - {	
104 TriLEDGpio_Put(3E,GPIO_LOW);	<pre>// Toggling Red led</pre>
105 Delay(1000);	// delay
106 TriLEDGpio_Put(3E,GPIO_HIGH);	
107 Delay(1000);	
108 TriLEDGpio_Put(3F,GPIO_LOW);	<pre>// Toggling Green led</pre>
109 Delay(1000);	
<pre>110 TriLEDGpio_Put(3F,GPIO_HIGH);</pre>	
111 Delay(1000);	
112 TriLEDGpio_Put(41,GPIO_LOW);	<pre>// Toggling Blue led</pre>
113 Delay(1000);	
114 TriLEDGpio_Put(41,GPIO_HIGH);	
115 Delay(1000);	
116 }	
117 -	
118 }	
119	

Figure 15: TriLED\_App Function

- 5) TriLED\_App() Function:
  - a. The TriLEDGpio\_Put function is used to change the value assigned to the pins. GPIO\_LOW indicates LED On and GPIO\_HIGH indicates LED Off.



- 6) Configure Keil for WyzBee<sup>™</sup> as explained in Section 3.2.
- 7) Compile the project by clicking on Project  $\rightarrow$  Build Target.
- 8) After successful compilation, click on Flash  $\rightarrow$  Erase to Erase the MCU's Flash contents.
- Next, click on Debug → Start/Stop Debug Session to download the application to the MCU. The IDE now shows the debug cursor pointing to the main function in the main.c file.



#### Figure 16: Debug Cursor at main Function

10) Use the Run (Ctrl+F5), Step-in (F11), Step-out (F10) and Step-over (Ctrl+F11) options to execute and debug the application.

For more details on how to use Keil uVision4 IDE, refer the document from the following link:

#### http://www.keil.com/product/brochures/uv4.pdf

You can now start writing your own Application using the API libraries provided. The APIs are explained in WyzBee API Guide document.

## 3.4 Adding THINGS

The WyzBee<sup>™</sup> THING expansion headers accommodate a host of other symbiotic devices, with a number of peripherals – called 'Things' – already available from Redpine including audio, GSM, GPS, capacitive touch display, rechargeable battery, and additional sensors.

Adding THINGs is as simple as stacking one board on top of the other. Care needs to be taken to ensure that the interfaces being used for one THING are not being used by another THING when stacking multiple THINGs.

Details on the THINGs are available in their individual documents.



## 4 Appendix A: Installing IDEs

This section describes the steps for downloading and installing the  $IDEs^1$  which can be used with WyzBee<sup>M</sup>.

## 4.1 Keil IDE

The Keil IDE from ARM can be used for IoT application development and supports both C and C++ languages.

#### 4.1.1 Download and Installation

Evaluation version of Keil IDE can be downloaded from <u>https://www.keil.com</u>.The following steps have been described for Keil version 4.7.4.

We strongly recommend ensuring you are downloading the MDK-ARM V4.74 setup file.

You need to register yourself to generate the setup file to be downloaded.

- 1) Start the installer. If any security warnings appear, click on Run.
- 2) In the Installer window, click Next.
- 3) Click Next again to continue the installation.

Setup MDK-ARM V4.74	
Welcome to Keil MDK-ARM Release 3/2014	
This SETUP program installs:	
MDK-ARM V4.74	
This SETUP program may be used to update a However, you should make a backup copy be	a previous product installation. fore proceeding.
It is recommended that you exit all Windows p	ograms before continuing with SETUP.
Follow the instructions to complete the produc	t installation.
— Keil MDK-ARM Setup ————	
	<< Back Next >> Cancel

## Figure 17: Keil IDE Installation Window

4) Check the "I agree to all the terms..." option and click Next.

<sup>&</sup>lt;sup>1</sup> Keil and IAR impose a code size limitation of 32KB on the evaluation versions of their IDEs





icense Agreement		> KEIL
Please read the rollowing license agreement of	sareruliy.	Tools by ARM
To continue with SETUP, you must accept the agreement, click the check box below.	e terms of the License Agreeme	nt. To accept the
END USER LICENCE AGREEMENT FO THIS END USER LICENCE AGREEM BETWEEN YOU (EITHER A SINGLE IN ARM LIMITED ("ARM") FOR THE USE LICENCE. ARM IS ONLY WILLING T CONDITION THAT YOU ACCEPT AL CLICKING "I AGREE" OR BY INSTALL	R MDK-ARM ENT ("LICENCE") IS A LE IDIVIDUAL, OR SINGLE LE OF THE SOFTWARE ACC O LICENSE THE SOFTW L OF THE TERMS IN TI LING OR OTHERWISE US	GAL AGREEMENT GAL ENTITY) AND OMPANYING THIS (ARE TO YOU ON HIS LICENCE. BY SING OR COPYING -
F I agree to all the terms of the preceding Lic	cense Agreement	

#### Figure 18: Keil IDE Installation License Agreement

5) In the next window, you have the option of changing the folder in which the IDE will be installed. Change it if required and click Next.

Setup MDK-ARM V4.74
Folder Selection         Select the folder where SETUP will install files.
SETUP will install MDK-ARM in the following folder. To install to this folder, press 'Next'. To install to a different folder, press 'Browse' and select another folder.
Destination Folder Browse
- Keil MDK-ARM Setup

#### Figure 19: Keil IDE Installation Folder Selection

6) Enter your details in the next window and click Next to start the installation process.





Setup MDK-ARM V4.74		<b>•</b>
Customer Informati Please enter your in	on formation.	
Please enter your na	me, the name of the company for whom you wo	rk and your E-mail address.
First Name:		
Last Name:		
Company Name:	Redpine Signals	
E-mail:	redpinesignals.com	
— Keil MDK-ARM Setup	<< B.	ack Next>> Cancel

## Figure 20: Keil IDE Customer Information

7) After the installation is completed, click Next again.

		83
		Is by ARM
	•	
<< Back	Next>>	Cancel
	<< Back	<< Back Next>>

## Figure 21: Keil IDE Installation Completed

8) Now, select the "Launch Driver Installation: ULINK Pro Driver V1.0" option and click Finish.





Setup MDK-ARM V4.74	×
Keil MDK-ARM Setup completed MDK-ARM V4.74	
µVision Setup has performed all requested operations suc Launch Driver Installation: "ULINK Pro Driver V1 Show Release Notes.	:cessfully. 1.0"
— Keil MDK-ARM Setup ————————————————————————————————————	<< Back Finish Cancel

Figure 22: Keil IDE Installation Completed

9) In the new window that opens, click Install to continue with the installation of the ULINK drivers.



Figure 23: ULINK Driver Installation

## 4.1.2 Configuring Keil IDE for WyzBee™

The process for configuring Keil IDE for WyzBee<sup>™</sup> is explained in <u>Section 3.2</u>.

#### 4.2 ColDE

The CoIDE from CooCox is a free IDE and can be used for IoT application development. CoIDE supports both C and C++ languages and does not have any code size limitations.





#### 4.2.1 Download and Installation

CoIDE can be downloaded from <u>http://www.coocox.org</u>. The following steps have been described for CoIDE version 1.7.8.

- 1) Start the installer. If any security warnings appear, click on Run.
- 2) In the Installer window, click Next.



#### Figure 24: COIDE Installation Window

3) In the next window, you have the option of changing the folder in which the IDE will be installed. Change it if required and click Next.





Setup - CoIDE	
Select Destination Location	
Where should CoIDE be installed?	
Setup will install CoIDE into the following folder.	
To continue, click Next. If you would like to select a different	t folder, dick Browse.
C:\CooCox\CoIDE	Browse
	E.
At least 433,9 MB of free disk space is required.	

Figure 25: CoIDE Installation Folder Selection

4) In the new window, click Install to start the Installation of CoIDE.

Setup is now ready to begin inst	talling CoIDE on your computer.
Click Install to continue with the change any settings.	installation, or click Back if you want to review or
Destination location: C:\CooCox\CoIDE Start Menu folder: CooCox Software	*
٠	

Figure 26: COIDE Installation Start

5) Click Finish after the installation is completed.





- 6) CoIDE doesn't come with an integrated GCC compiler. Download and install the relevant GCC toolchain for Windows from <a href="https://launchpad.net/gcc-arm-embedded/">https://launchpad.net/gcc-arm-embedded/</a>
- 7) Once the installation of the GCC Toolchain is completed, open CoIDE and click on "Select Toolchain Path" under "Project". In the new dialog box that opens, enter the path where the GCC Toolchain was installed.

r select i doichain Path Toolchain Bath: ( the directon/where "arm-none-eahi-goo eve"	"evicted )
C:\Program Files\GNU Tools ARM Embedded\4.8 2014q2\bin	Browse

Figure 27: GCC Toolchain Path for CoIDE

8) To verify the downloaded version, click Help -> About CoIDE

🕕 About CoIDE		×
	CooCox CoIDE Version: 1.7.8 Build id: 20141023-3334-1557 CooCox CoIDE is a free software product. Visit: http://www.coocox.org/CooCox_CoIDE.htm	
	ОК	

#### Figure 28: CoIDE Version

## 4.2.2 Configuring CoIDE for WyzBee™

Follow the steps below to configure CoIDE for WyzBee<sup>™</sup>.

1) Open CoIDE and click on the Configuration button.

jec	t Fl	ash	Debug	Se	arch		Help				
*	*	8	Ø	(ĝ)	٩			毘	3	1000	R
			Courses .		1	1	10 P.			1	3

Figure 29: CoIDE Configuration Button

2) In the new window that opens, ensure that the settings on each tab match the settings in the images below.





## Configuration

evice	Compile	Link	Output	User	Debugger	Download
evice						
type	filter text Spansic MB	n 98560R_	N_M	A S	anufacturer: eries: evice: polset:	Spansion MB9B560R_N_M MB9BF568NBGL ARM
		MB9BF5 MB9BF5 MB9BF5 <mark>MB9BF5</mark>	668RPMC 668RBGL 668NPMC 668NBGL	M	IB9BF568NBG escriptions:	L



evice Co	mpile Link Output User	Debugger Download			
ink					
Options			Linked Li	braries	
🔲 Use Me	emory Layout from Memory Wi	ndow			Ad
V Discard	l unused sections				
📃 Don't u	ise the standard system startup	files			Kem
Enable	LTO				
Library:	Not use C Library 🔹				
	· · · ·				
Memory /	Areas				
Read/on	ly Memory Area		Read/wr	ite Memory Area	
on-chip	Start	Size	on-chip	Start	Size
IROM1	0x0000000	0x00100000	IRAM1	0x1FFF0000	0x00010000
IROM2	0x200C0000	0x00008000	IRAM2	0x20038000	0x00010000
	in Elash 🖉 Debug in RAM				
@ Debug	, and a set of a set of a set of a				
Debug					
② Debug Scatter Fil	e				
Debug Scatter Fil ,/arm-gc	e c-link.ld				
Debug Scatter Fil ,/arm-gc	e c-link.ld				
Debug     Scatter Fil     ./arm-gc     Misc Cont	e c-link.ld trols				
Debug Scatter Fil ,/arm-gc Misc Cont -lstdc++	e c-link.ld trols				

Figure 31: CoIDE Configuration – Link





## Configuration

Device Compile Link	k Output User Debugger	Download	
Output			
Options			
Name of Executable:	TriLed	Path of Executable:	./
Oreate Executable	(.elf):		
Create HEX F	File		
🔽 Create BIN F	île		
Create Library(.a)			

## Figure 32: CoIDE Configuration – Output

Device Compile Link Output User D	ebugger Download		
User			
Buildin variables			
Before Build/Rebuild		After Build/Rebuild	
Run#1		Run#1	
Run#2		Run#2	
Before Debug			
Auto Compile Before Debugging			

#### Figure 33: CoIDE Configuration – User

Con	fig	ura	tio	n
-----	-----	-----	-----	---

Device	Compile	Link	Output	User	Debugger	Download			
Debug	jer								
Hard	vare								
Adapt	er CMSI	S-DAP				•	▼ Max Clock(H	z) [1M	•]
Startu	p								
🔽 Ru	n to main								
Adva	nce								
Reset	Mode SY:	SRESETF	REQ 👻	Se Se	mihosting En	able [			
Targe	tInfo								
Host	P Address:	127	. 0	. 0	. 1		Port Number:	2009	

Figure 34: CoIDE Configuration – Debugger





Co	nfi	au	rat	ion
		9-		

Device	Compile	Link	Output	User	Debugger	Download
DownL	oad	14	alla.			
Down	nload					
V Au	uto Downlo	ad Befo	re Debuggi	ng		
Verif	/					
Ve	rify After D	ownloa	d			
Enner						
Erase	E 11 CL 1					
C Er	ase Full Chi	p 🧕	Erase Effe	cted	O Do not Er	ase
Prog	ramming A	lgorithn	n			
file	Path					
C:/(	CooCox/Co	IDE/flas	h/mb9bfxx	8_1024.e	elf	
Add	Rem	ove	default			
Aud		ove	ucrauit			

Figure 35: CoIDE Configuration – Download

## 4.3 IAR Embedded Workbench

IAR Embedded Workbench is a development environment that includes a C/C++ compiler and debugger.

#### 4.3.1 Download and Installation

The IAR Embedded Workbench can be downloaded from <u>https://www.iar.com</u>. The following steps have been described for IAR version 7.30.

- 1) Start the installer. If any security warnings appear, click on Run.
- 2) In the installation window, selection the "Install IAR Embedded Workbench" option.





IAR Embedded Workbench®	<ul> <li>Japanese</li> </ul>
Installation and licensing information	
Install IAR Embedded Workbench <sup>™</sup> <del>←</del> Release notes Install drivers	
Explore the installation media Exit	
-0-0-0	<b>EIAR</b> SYSTEMS www.iar.com

#### Figure 36: IAR Embedded Workbench Installation Window

3) In the new window, click Next.



## Figure 37: IAR Embedded Workbench Installation Window

4) Check the "I accept the terms of the license agreement" option and click Next.





Pleas	e read the following	license agreemen	it carefully.		
		- muni			SYS
SOF1 Conc Syste	WARE LICENSE A erning the Products ms AB	AGREEMENT : IAR Embedded \	Workbench® and	visualSTATE®	from IAR
PREA	AMBLE				
THIS PROI SUBS YOU	SOFTWARE LICE DUCTS LICENSED SIDIARIES ("IAR S" ("THE LICENSEE"	NSE AGREEMEN BY IAR SYSTEM YSTEMS'') OR DI ).	T ("THE AGREEN S AB INCLUDING STRIBUTORS ("I/	IENT'') APPLIE ANY OF ITS AR DISTRIBUT	IS TO TORS''), TO
<b>T</b> 1	1 e e 1	MD C I	- i e ii	a	
📕 🔍 la	ccept the terms of t	he license agreem	nent		Print
© I d	o not accept the te	rms of the license	agreement		
nstallShiel	1				

#### Figure 38: IAR Embedded Workbench License Agreement

5) In the next window, you have the option of changing the folder in which the IDE will be installed. Change it if required and click Next.



## Figure 39: IAR Embedded Workbench Installation Folder





6) Click Next in the new window and then click Install to start the installation process.

IAR Embedded Workbench for ARM 7.30.4	<b></b>
Ready to Install the Program The wizard is ready to begin installation.	<b>SYSTEMS</b>
Click Install to begin the installation.	
If you want to review or change any of your instal the wizard.	llation settings, click Back. Click Cancel to exit
InstallShield	< Back Install Cancel

#### Figure 40: IAR Embedded Workbench Installation Start

7) Wait for the installation to be completed. You will be requested to remove any USB dongles connected to the PC. Remove them and click Yes.



#### Figure 41: IAR Embedded Workbench – Dongle Drivers Installation

8) This will start the installation of the dongle drivers. Once completed, select the "Launch IAR Embedded Workbench for ARM" option and click Finish to complete the installation of IAR Embedded Workbench.





IAR Embedded Workbench for	ARM 7.30.4
-11- 11-	InstallShield Wizard Complete
VIIII VIIII VIIII VIIII	The InstallShield Wizard has successfully installed IAR Embedded Workbench for ARM. Click Finish to exit the wizard.
	View the release notes
	Launch IAR Embedded Workbench for ARM
<b>EIAR</b> SYSTEMS	
	< Back Finish Cancel

## Figure 42: IAR Embedded Workbench Installation Completed

9) Next, you will be prompted for installation of device software. Click Install.



#### Figure 43: IAR Embedded Workbench – Device Software – 1







## Figure 44: IAR Embedded Workbench – Device Software – 2

10) Select the Language of your preference and click OK.

anguage	<b>X</b>
Select the language for the IAR Embedded	. ОК
English (United States)	•

Figure 45: IAR Embedded Workbench – Language Selection

11) The IAR Embedded Workbench window opens, followed by the License Wizard. Click Next.



Figure 46: IAR Embedded Workbench – License Wizard

12) Click Next again.





License Wizard	<b>×</b>
Choose a product	
Select the product you want to evaluate.	
IAR Embedded Workbench for ARM	
	<b>IAR</b>
	SYSTEMS
	< Back Next > Cancel

## Figure 47: IAR Embedded Workbench – License Wizard (Choose a Product)

13) In the new window, click Register.

License Wizard	×
Register	
When you register you will receive a license number for an evaluation license.	
Register	
Enter the license number you received after registering and click Next.	
	<b>R</b> MS
< <u>B</u> ack <u>N</u> ext > (	Cancel

## Figure 48: IAR Embedded Workbench – License Wizard (Register)





14) Clicking Register opens a web browser for the Registration process. Select the "Code Size Limited" option and enter the rest of the details.

Eval	uation license type *
0	Time limited (30 days) IAR Embedded Workbench for ARM, v. 7.30, Evaluation Edition
0	Code size limited IAR Embedded Workbench for ARM, v. 7.30, 32K Kickstart Edition
First	name *
Last	name *

#### Figure 49: IAR Embedded Workbench: Registration

- 15) Once the details are submitted, a confirmation email is sent to the registered email address. Open the confirmation email and click on the confirmation link.
- 16) Next, a page opens with a License Key.





<b>IAR</b> SYSTEMS	English • 日本語
Registration Complete	
Thank you for your registration!	
You have been assigned the following license number:	
IAR Systems	
IAR Systems website	

#### Figure 50: IAR Embedded Workbench: Registration Complete

17) Copy this key and paste it in the License Wizard and click Next to complete the installation and registration process for IAR Embedded Workbench.

#### 4.3.2 Configuration IAR Embedded Workbench for WyzBee™

1) Open the IAR Embedded Workbench and click on Project  $\rightarrow$  Options



Figure 51: IAR Embedded Workbench – Project Options





2) In the Options window that opens, ensure that all settings are as per the images shown below.

Category:		Factory Setting
Seneral Options		
C/C++ Compiler		
Assembler	Salua Charles and Energy	
Output Converter		ons   Plugins
Custom Build		
Build Actions		un to
Linker	J-Link/J-Trace	ain
Debugger	Simulator	
Simulator	Angel	
Angel	CMSIS DAP (3)	
CMSIS DAP	LAP POM monitor	
GDB Server	Liet/ITAGiet	
IAR ROM-monitor	J-Link/J-Trace	
I-jet/JTAGjet	TI Stellaris	
J-Link/J-Trace	Macraigor	
TI Stellaris	PE micro	
Macraigor	ST-LINK	
PE micro	Third-Party Driver	
RDI	TI XDS100	
ST-LINK		
Third-Party Driver	1	
TT VDS100		

Figure 52: IAR Embedded Workbench – Options – 1

General Options C/C++ Compiler			Factory Setting
Assembler Output Converter	JTAG/SWD	Breakpoints	
Custom Build 7	Sterface Pr	obe configuration	
Build Actions		Multi-target debug system	
Linker		Target number (TAP or Multidrop ID):	
Simulator	(b)	Terraet with multiple CPUs	
ingel		COLUMN AND AND AND AND AND AND AND AND AND AN	-
CMSIS DAP	ITAG/SW/D speed	CPO number on target.	
GDB Server	Auto detect		
IAR ROM-monitor	Auto detect		
J-Link/J-Trace			
TI Stellaris			
Macraigor			
PE micro			
ST-LINK			

Figure 53: IAR Embedded Workbench – Options – 2

3) Click OK to complete the configuration for WyzBee<sup>™</sup>.





## 5 Appendix B: WyzBee<sup>™</sup> Headers

The WyzBee<sup>™</sup> baseboard comes with 2 16-pin THING<sup>™</sup> Headers. These headers can be used to add a combination of multiple THING<sup>™</sup> boards.

INT08_2S0T6_0/SDA6_0P56S1	S17	VDD_3V3				
RT015_0 INT07_2 SIN6_0 P55 S2		P3D RT003_0 TIOA3_1				
TIOB1_1 INT09_2 SIN3_1 P31 S3		P4C SCK7_1/SCL7_1 AIN1_2	TIOB2_0			
TIOB3_1 INT04_1 SCK3_1/SCL3_1 P33 S4		P4E SIN7_1 INT11_1	TIOB4_0 ZIN1_2 WKUP2			
TIOB2_1 INT10_1 SOT3_1/SDA3_1 P32 S5		P4B SCS7_1 TIOB1_0				
TIOA1_1 RTO01_0 BIN0_0 P3B S6		P4D SOT7_1/SDA7_1 INT13_2	TIOB3_0 BIN1_2			
TIOA2_1 RTO02_0 ZIN0_0 P3C S7						
MCU_RESET S8						
TIOA7_1 RT000_1 AN15 P23 S9	S24	P15 SCK0_1/SCL0_1 AN05				
TIOA4_0 RT014_1 DAC0 P44 S10	Savia 8	VDD_MOD				
AN06 INT14_1 SIN2_2 P16 S11	S26	P3A TIOA0_1 AIN0_0	RT000_0			
WKUP3 AN07 SOT2_2 P17 S12		P14 SOT0_1/SDA0_1 AN04	IC03_0			
AN08 SCK2_2/SCL2_2 P18 S13		P13 SIN0_1 AN03	IC02_2 INT03_1			
VDD_5V S14		P12 SCK1_1/SCL1_1 AN02	IC01_2			
GND S15	S30 - S30 -	P11 SOT1_1/SDA1_1 AN01	IC00_2 TX1_2			
VIN_EXT SIG SINI_1 AN00 INTO2_1 RX1_2						
General Purpose Input/Output	e/Multi-Function Timer	QPRC	Power/Ground			
Multi-Function Serial Interface ADC	C/DAC	CAN Interface				
External Interrupt Input Capture Reset/Wakeup						

Figure 54: 32-pin WyzBee™ THING Header

## 5.1 Pin Description

The table below describes the pins of the WyzBee<sup>™</sup> baseboard's Headers.

Pin Name	Function	Direction	Description		
	THING Headers				
S1	P56	Inout	General-purpose Input/Output		
	SOT6_0/SDA6_0	Output/Inout	Multi-function Serial Interface Channel 6 Output/Inout		
	INT08_2	Input	External Interrupt 8		
S2	P55	Inout	General-purpose Input/Output		
	SIN6_0	Input	Multi-function Serial interface Channel 6 Input		
	RTO15_0	Output	Waveform generator output of Multi-function timer 15		





Pin Name	Function	Direction	Description
	INT07_2	Input	External Interrupt 7
S3	P31	Inout	General-purpose Input/Output
	TIOB1_1	Inout	Base Timer Channel 1 TIOB pin. This can be used only if S16 is not configured as TIOB1_0.
	SIN3_1	Input	Multi-function Serial interface Channel 3 Input
	INT09_2	Input	External Interrupt 9
S4	P33	Inout	General-purpose Input/Output
	TIOB3_1	Inout	Base Timer Channel 3 TIOB pin. This can be used only if S17 is not configured as TIOB3_0.
	SCK3_1/SCL3_1	Output	Multi-function Serial interface Chanel 3 Clock output
	INT04_1	Input	External Interrupt 4
S5	P32	Inout	General-purpose Input/Output
	TIOB2_1	Inout	Base Timer Channel 2 TIOB pin. This can be used only if S14 is not configured as TIOB2_0.
	SOT3_1/SDA3_1	Output/Inout	Multi-function Serial Interface Channel 3 Output/Inout
	INT10_1	Input	External Interrupt 10
S6	РЗВ	Inout	General-purpose Input/Output
	TIOA1_1	Inout	Base Timer Channel 1 TIOA pin
	BINO_0	Input	QPRC Channel 0 BIN Input
	RTO01_0	Output	Waveform generator output of Multi-function timer 1
S7	РЗС	Inout	General-purpose Input/Output
	TIOA2_1	Inout	Base Timer Channel 2 TIOA pin





Pin Name	Function	Direction	Description
	ZINO_0	Input	QPRC Channel 0 ZIN Input
	RTO02_0	Output	Waveform generator output of Multi-function timer 2
S8	MCU_RESET_N	Input	External reset
S9	P23	Inout	General-purpose Input/Output
	TIOA7_1	Inout	Base Timer Channel 7 TIOA pin
	RTO00_1	Output	Waveform generator output of Multi-function timer 0
	AN15	Input	ADC Input Channel 15
S10	P44	Inout	General-purpose Input/Output
	TIOA4_0	Inout	Base Timer Channel 4 TIOA pin
	RTO14_1	Output	Waveform generator output of Multi-function timer 14
	DACO	Output	DAC Output Channel 0
S11	P16	Inout	General-purpose Input/Output
	SIN2_2	Input	Multi-function Serial interface Channel 2 Input
	INT14_1	Input	External Interrupt 14
	AN06	Input	ADC Input Channel 6
S12	P17	Inout	General-purpose Input/Output
	SOT2_2	Output	Multi-function Serial Interface Channel 2 Output
	AN07	Input	ADC Input Channel 7
	WKUP3	Input	Deep standby mode return Channel 3 Input
S13	P18	Inout	General-purpose Input/Output
	SCK2_2/SCL2_2	Output	Multi-function Serial interface





Pin Name	Function	Direction	Description
			Chanel 2 Clock output
	AN08	Input	ADC Input Channel 8
S14	VDD_5V	Output	5V Output Power Supply for THING boards.
S15	GND	Ground	Ground
S16	VIN_EXT	Input	External Power Supply. This can be from different sources like battery THING.
S12	P15	Inout	General-purpose Input/Output
	SCK0_1/SCL0_1	Output	Multi-function Serial interface Chanel 0 Clock output
	AN05	Input	ADC Input Channel 5
S17	VDD_3V3	Output	3.3V Output Power supply
S18	P4C	Inout	General-purpose Input/Output
	TIOB2_0	Inout	Base Timer Channel 2 TIOB pin
	SCK7_1/SCL7_1	Output	Multi-function Serial interface Chanel 7 Clock output
	AIN1_2	Input	QPRC Channel 1 AIN Input
S19	P4E	Inout	General-purpose Input/Output
	TIOB4_0	Inout	Base Timer Channel 4 TIOB pin
	SIN7_1	Input	Multi-function Serial interface Channel 7 Input
	ZIN1_2	Input	QPRC Channel 1 ZIN Input
	INT11_1	Input	External Interrupt 11
	WKUP2	Input	Deep standby mode return Channel 2 Input
S20	P4B	Inout	General-purpose Input/Output





Pin Name	Function	Direction	Description
	TIOB1_0	Inout	Base Timer Channel 1 TIOB pin. This can be used only if S12 is not configured as TIOB1_1.
	SCS7_1	Output	Multi-function Serial interface Channel 7 Chip Select
S21	P4D	Inout	General-purpose Input/Output
	TIOB3_0	Inout	Base Timer Channel 3 TIOB pin. This can be used only if S3 is not configured as TIOB3_0.
	SOT7_1/SDA7_1	Output/Inout	Multi-function Serial Interface Channel 2 Output/Inout
	BIN1_2	Input	QPRC Channel 1 BIN Input
	INT13_2	Input	External Interrupt 13
S22	Ground	Ground	Ground
S23	MCU_VCC33	Output	3.3V Power Supply
S24	P3D	Inout	General-purpose Input/Output
	TIOA3_1	Inout	Base Timer Channel 3 TIOA pin
	RTO03_0	Output	Waveform generator output of Multi-function timer 3
S25	VDD_MOD	Output	3.3V Output Power supply
S26	РЗА	Inout	General-purpose Input/Output
	TIOA0_1	Inout	Base Timer Channel 0 TIOA pin
	AIN0_0	Input	QPRC Channel 0 AIN Input
	RTO00_0	Inout	Waveform generator output of Multi-function timer 0
S27	P14	Inout	General-purpose Input/Output
	SOT0_1/SDA0_1	Output/Inout	Multi-function Serial Interface Channel 0 Output/Inout





Pin Name	Function	Direction	Description
	IC03_2	Input	16-bit input capture Channel 3 input pin of Multi-function timer 0
	AN04	Input	ADC Input Channel 4
S28	P13	Inout	General-purpose Input/Output
	SIN0_1	Input	Multi-function Serial interface Channel 0 Input
	IC02_2	Input	16-bit input capture Channel 2 input pin of Multi-function timer 0
	INT03_1		External Interrupt 3
	AN03	Input	ADC Input Channel 3
S29	P12	Inout	General-purpose Input/Output
	SCK1_1/SCL1_1	Output	Multi-function Serial interface Chanel 1 Clock output
	IC01_2	Input	16-bit input capture Channel 1 input pin of Multi-function timer 0
	RTCC0_1	Output	Reserved
	AN02	Input	ADC Input Channel 2
S30	P11	Inout	General-purpose Input/Output
	TX1_2	Output	CAN interface Channel 1 TX output
	SOT1_1/SDA1_1	Output/Inout	Multi-function Serial Interface Channel 1 Output/Inout
	IC00_2	Input	16-bit input capture Channel 0 input pin of Multi-function timer 0
	AN01	Input	ADC Input Channel 1





Pin Name	Function	Direction	Description
S31	P10	Inout	General-purpose Input/Output
	INT02_1	Input	External Interrupt 2
	RX1_2	Output	CAN interface Channel 1 RX input
	SIN1_1	Input	Multi-function Serial interface Channel 1 Input
	AN00	Input	ADC Input Channel 0

Table 2: Headers Pins Description

\*\*\*\*

# **Mouser Electronics**

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

**Redpine Signals:** 

WyzBee-BASE-001 WyzBee-SENS-101 WyzBee-DISP-108 WyzBee-SADK-901 WyzBee-BATT-107