Navigating a Processor's Reference Manual

Overview

Even though our project mostly utilizes pre-developed drivers for ADC, GPIO, CAN, etc... peripherals for our RC car project, we still need to have a good understanding of their underlying usage. This is even more important if we need to debug any integration issues when connecting new sensors or components as we build the car. If we end up using a new peripheral for a component, we need to know how to write our own driver for it. At that point, we need to navigate our processor's reference manual to learn the peripheral's functionality and configuration.

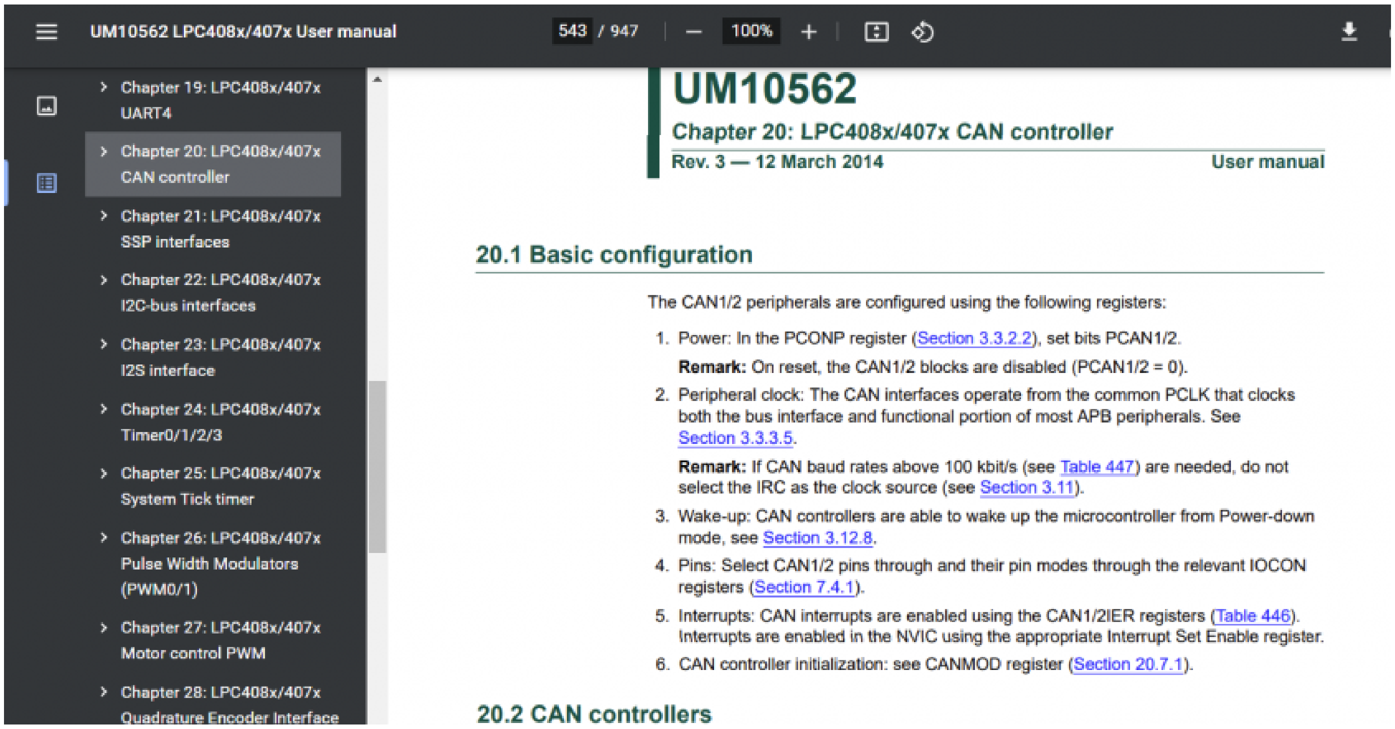
Reference Manual vs Datasheet

LPC408x Reference Manual: Reference Manual

The Reference Manual contains all functional and usage descriptions of the processor series and its peripherals. The Datasheet describes the mechanical and electrical characteristics of the specific processor model. For our purposes, we can think of using the Reference Manual to learn about a certain peripheral (UART, CAN, etc...) and how to configure it, while using the Datasheet to figure out what external pins we can use to connect to the peripheral.

How to Find the Details of a Peripheral You Are Working With

The reference manual is a huge document, so we always need to have a specific peripheral in mind and start with the **table of contents**. For our example, we will use the CAN peripheral. We look for the chapter that contains the "CAN Controller". This same process works for any other peripheral like UART, PWM, etc...


UM10562 LPC408x/407x User manual
543 / 947
100%
Chapter 19: LPC408x/407x UART4
Chapter 20: LPC408x/407x CAN controller
Chapter 21: LPC408x/407x SSP interfaces
Chapter 22: LPC408x/407x I2C-bus interfaces
Chapter 23: LPC408x/407x I2S interface
Chapter 24: LPC408x/407x Timer0/1/2/3
Chapter 25: LPC408x/407x System Tick timer
Chapter 26: LPC408x/407x Pulse Width Modulators (PWM0/1)
Chapter 27: LPC408x/407x Motor control PWM
Chapter 28: LPC408x/407x Quadrature Encoder Interface

UM10562

Chapter 20: LPC408x/407x CAN controller

Rev. 3 — 12 March 2014 User manual

20.1 Basic configuration

The CAN1/2 peripherals are configured using the following registers:

1. Power: In the PCONP register (Section 3.3.2.2), set bits PCAN1/2.

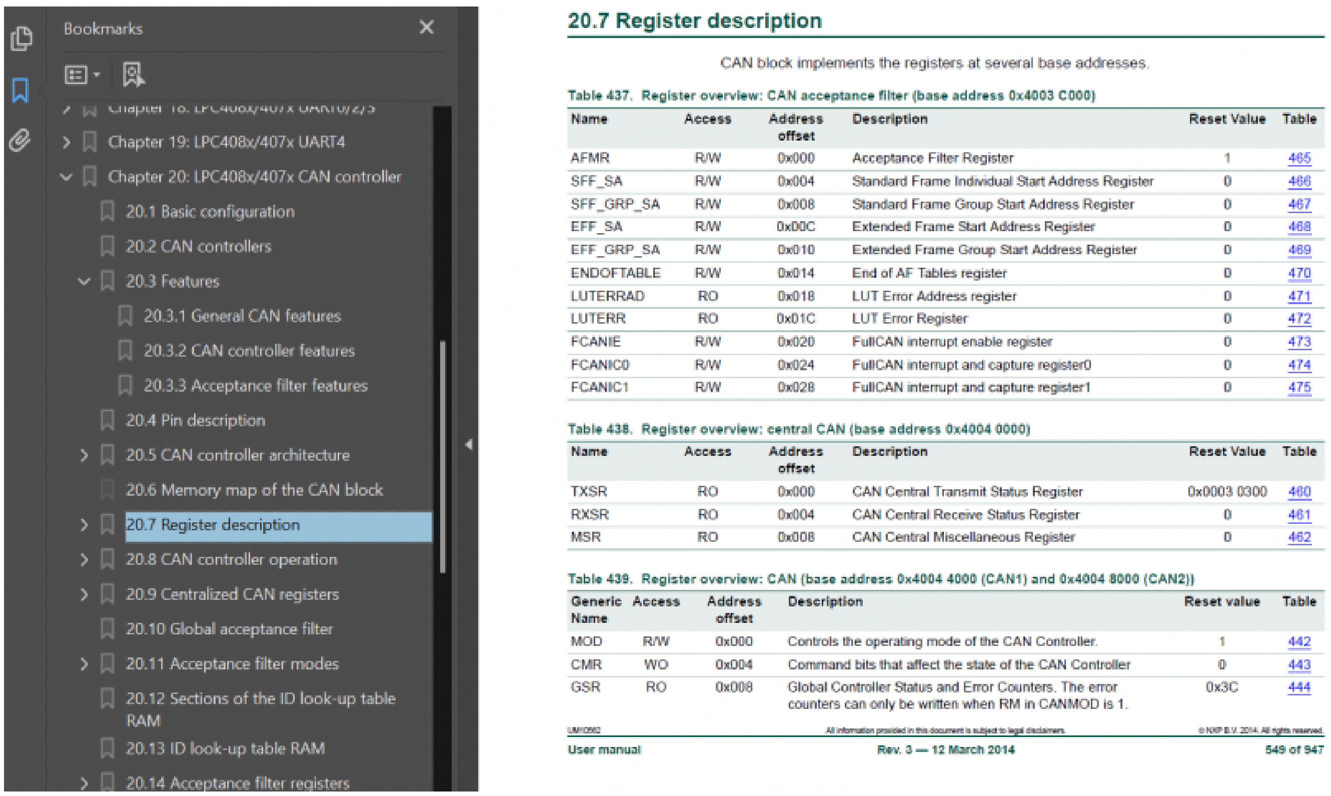
   Remark: On reset, the CAN1/2 blocks are disabled (PCAN1/2 = 0).
2. Peripheral clock: The CAN interfaces operate from the common PCLK that clocks both the bus interface and functional portion of most APB peripherals. See Section 3.3.3.5.

   Remark: If CAN baud rates above 100 kbit/s (see Table 447) are needed, do not select the IRC as the clock source (see Section 3.11).
3. Wake-up: CAN controllers are able to wake up the microcontroller from Power-down mode, see Section 3.12.8.
4. Pins: Select CAN1/2 pins through and their pin modes through the relevant IOCON registers (Section 7.4.1).
5. Interrupts: CAN interrupts are enabled using the CAN1/2IER registers (Table 446). Interrupts are enabled in the NVIC using the appropriate Interrupt Set Enable register.
6. CAN controller initialization: see CANMOD register (Section 20.7.1).

20.2 CAN controllers

Figure 1: Using the Table of Contents to find the specific peripheral

The Peripheral chapter contains its own informal table of contents with links to each section for configuration, functional description, and register details. We should read the functional description sections to get an understanding of how the peripheral could be used and any special functionalities. After understanding the peripheral, the next step is to learn about its registers and their configuration. Each peripheral chapter has a section for "Register description" which lists all registers.


Bookmarks
Chapter 19: LPC408x/407x UART4
Chapter 20: LPC408x/407x CAN controller
20.1 Basic configuration
20.2 CAN controllers
20.3 Features
20.3.1 General CAN features
20.3.2 CAN controller features
20.3.3 Acceptance filter features
20.4 Pin description
20.5 CAN controller architecture
20.6 Memory map of the CAN block
20.7 Register description
20.8 CAN controller operation
20.9 Centralized CAN registers
20.10 Global acceptance filter
20.11 Acceptance filter modes
20.12 Sections of the ID look-up table RAM
20.13 ID look-up table RAM
20.14 Acceptance filter registers

20.7 Register description

CAN block implements the registers at several base addresses.

Table 437. Register overview: CAN acceptance filter (base address 0x4003 C000)

Name	Access	Address offset	Description	Reset Value	Table
AFMR	R/W	0x000	Acceptance Filter Register	1	465
SFF_SA	R/W	0x004	Standard Frame Individual Start Address Register	0	466
SFF_GRP_SA	R/W	0x008	Standard Frame Group Start Address Register	0	467
EFF_SA	R/W	0x00C	Extended Frame Start Address Register	0	468
EFF_GRP_SA	R/W	0x010	Extended Frame Group Start Address Register	0	469
ENDOFTABLE	R/W	0x014	End of AF Tables register	0	470
LUTERRAD	RO	0x018	LUT Error Address register	0	471
LUTERR	RO	0x01C	LUT Error Register	0	472
FCANIE	R/W	0x020	FullCAN interrupt enable register	0	473
FCANIC0	R/W	0x024	FullCAN interrupt and capture register0	0	474
FCANIC1	R/W	0x028	FullCAN interrupt and capture register1	0	475

Table 438. Register overview: central CAN (base address 0x4004 0000)

Name	Access	Address offset	Description	Reset Value	Table
TXSR	RO	0x000	CAN Central Transmit Status Register	0x0003 0300	460
RXSR	RO	0x004	CAN Central Receive Status Register	0	461
MSR	RO	0x008	CAN Central Miscellaneous Register	0	462

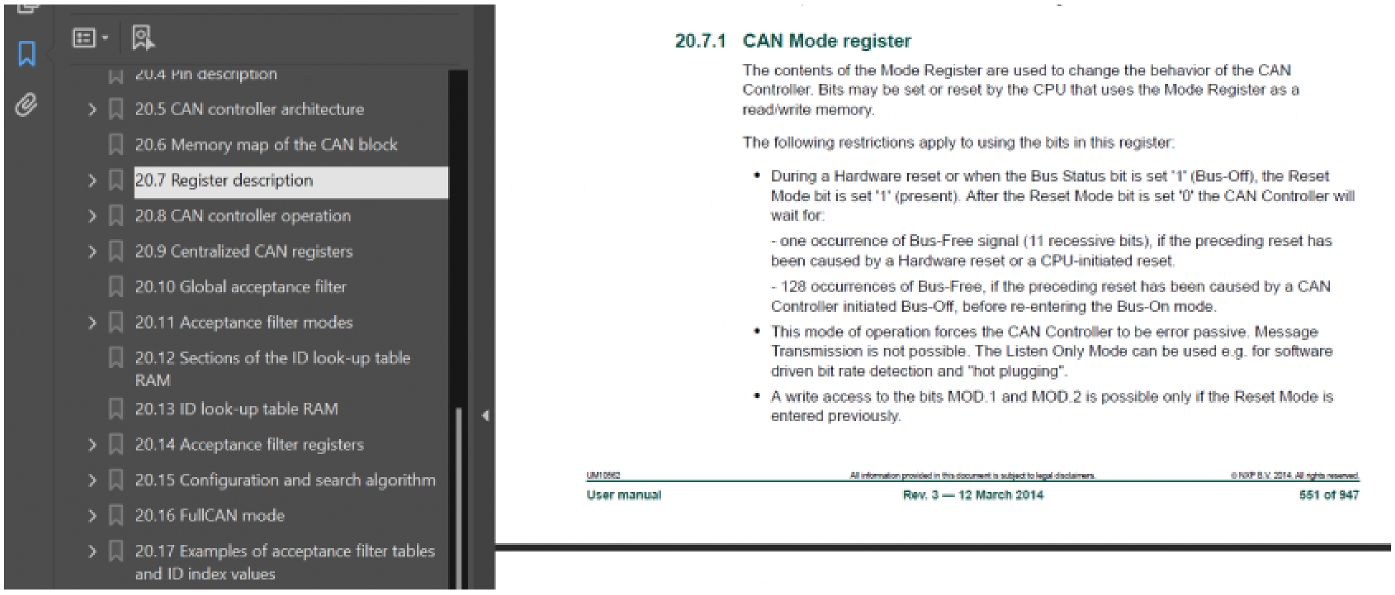
Table 439. Register overview: CAN (base address 0x4004 4000 (CAN1) and 0x4004 8000 (CAN2))

Generic Name	Access	Address offset	Description	Reset value	Table
MOD	R/W	0x000	Controls the operating mode of the CAN Controller.	1	442
CMR	WO	0x004	Command bits that affect the state of the CAN Controller	0	443
GSR	RO	0x008	Global Controller Status and Error Counters. The error counters can only be written when RM in CANMOD is 1.	0x3C	444

UM10562 All information provided in this document is subject to legal disclaimers. © NXP B.V. 2014. All rights reserved.
User manual Rev. 3 — 12 March 2014 549 of 947

Figure 2: The Register list section of the CAN peripheral chapter

From this list, we can link directly to a specific register you need to use. There will be a description of the controls contained within that specific register.


20.4 Pin description
20.5 CAN controller architecture
20.6 Memory map of the CAN block
20.7 Register description
20.8 CAN controller operation
20.9 Centralized CAN registers
20.10 Global acceptance filter
20.11 Acceptance filter modes
20.12 Sections of the ID look-up table RAM
20.13 ID look-up table RAM
20.14 Acceptance filter registers
20.15 Configuration and search algorithm
20.16 FullCAN mode
20.17 Examples of acceptance filter tables and ID index values

20.7.1 CAN Mode register

The contents of the Mode Register are used to change the behavior of the CAN Controller. Bits may be set or reset by the CPU that uses the Mode Register as a read/write memory.

The following restrictions apply to using the bits in this register:

- During a Hardware reset or when the Bus Status bit is set '1' (Bus-Off), the Reset Mode bit is set '1' (present). After the Reset Mode bit is set '0' the CAN Controller will wait for:

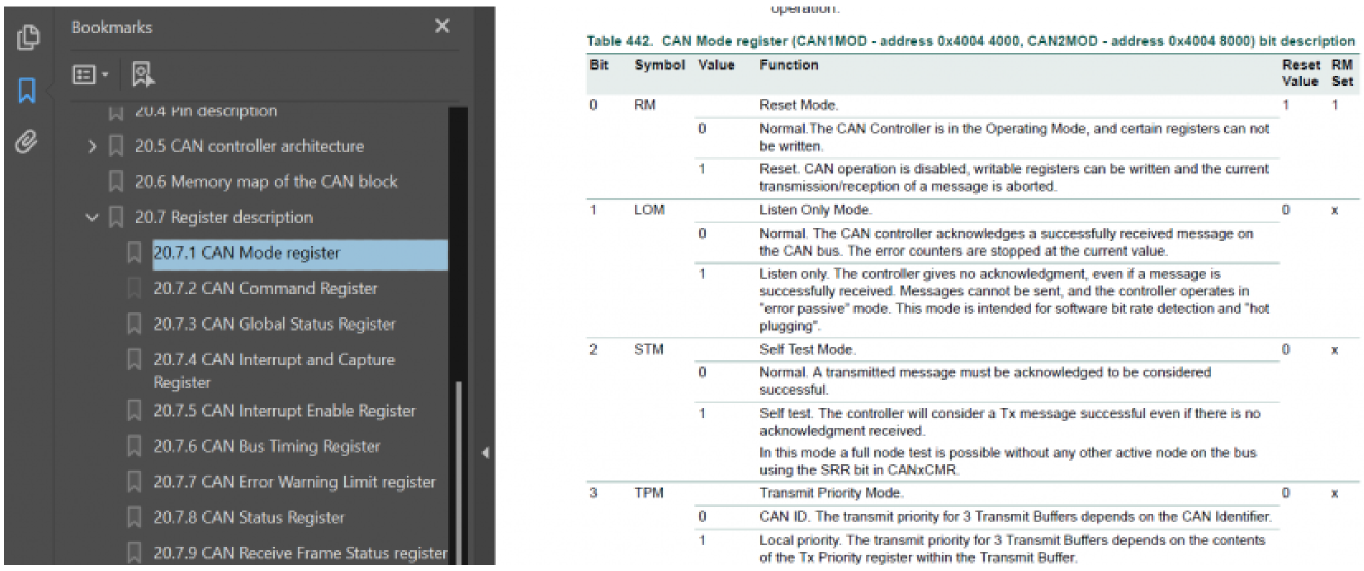
  - one occurrence of Bus-Free signal (11 recessive bits), if the preceding reset has been caused by a Hardware reset or a CPU-initiated reset.

  - 128 occurrences of Bus-Free, if the preceding reset has been caused by a CAN Controller initiated Bus-Off, before re-entering the Bus-On mode.
- This mode of operation forces the CAN Controller to be error passive. Message Transmission is not possible. The Listen Only Mode can be used e.g. for software driven bit rate detection and "hot plugging".
- A write access to the bits MOD.1 and MOD.2 is possible only if the Reset Mode is entered previously.

UM10562 All information provided in this document is subject to legal disclaimers. © NXP B.V. 2014. All rights reserved.
User manual Rev. 3 — 12 March 2014 551 of 947

Figure 3: Description of the CAN1MOD register and any relevant considerations

After the description, we will find the list of all control bits inside that specific register, along with their usage description.



operation.

Table 442. CAN Mode register (CAN1MOD - address 0x4004 4000, CAN2MOD - address 0x4004 8000) bit description

Bit	Symbol	Value	Function	Reset Value	RM Set
0	RM		Reset Mode.	1	1
		0	Normal.The CAN Controller is in the Operating Mode, and certain registers can not be written.		
		1	Reset. CAN operation is disabled, writable registers can be written and the current transmission/reception of a message is aborted.		
1	LOM		Listen Only Mode.	0	x
		0	Normal. The CAN controller acknowledges a successfully received message on the CAN bus. The error counters are stopped at the current value.		
		1	Listen only. The controller gives no acknowledgment, even if a message is successfully received. Messages cannot be sent, and the controller operates in "error passive" mode. This mode is intended for software bit rate detection and "hot plugging".		
2	STM		Self Test Mode.	0	x
		0	Normal. A transmitted message must be acknowledged to be considered successful.		
		1	Self test. The controller will consider a Tx message successful even if there is no acknowledgment received. In this mode a full node test is possible without any other active node on the bus using the SRR bit in CANxCMR.		
3	TPM		Transmit Priority Mode.	0	x
		0	CAN ID. The transmit priority for 3 Transmit Buffers depends on the CAN Identifier.		
		1	Local priority. The transmit priority for 3 Transmit Buffers depends on the contents of the Tx Priority register within the Transmit Buffer.		

Figure 4: Control bit descriptions for the entire CAN1MOD register

Using the Datasheet to Find Pin Outputs

LPC408x Datasheet: Datasheet

The last step when using a peripheral is to determine which GPIO pins can be used to connect to an external circuit or component. The datasheet (separate document) describes each pin connection and its available usages on our specific processor model. In this example, we see that the CAN1 receiver input signal is connected to the P0[0] (labeled P0.0 on the SJTwo board) GPIO pin.

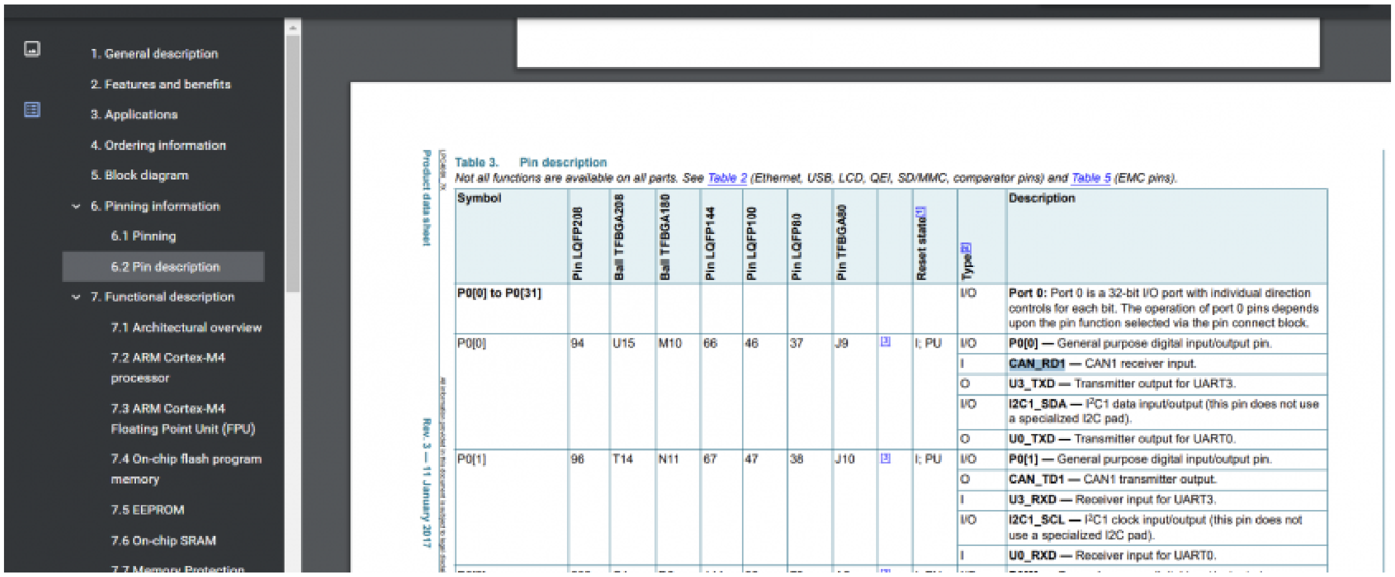
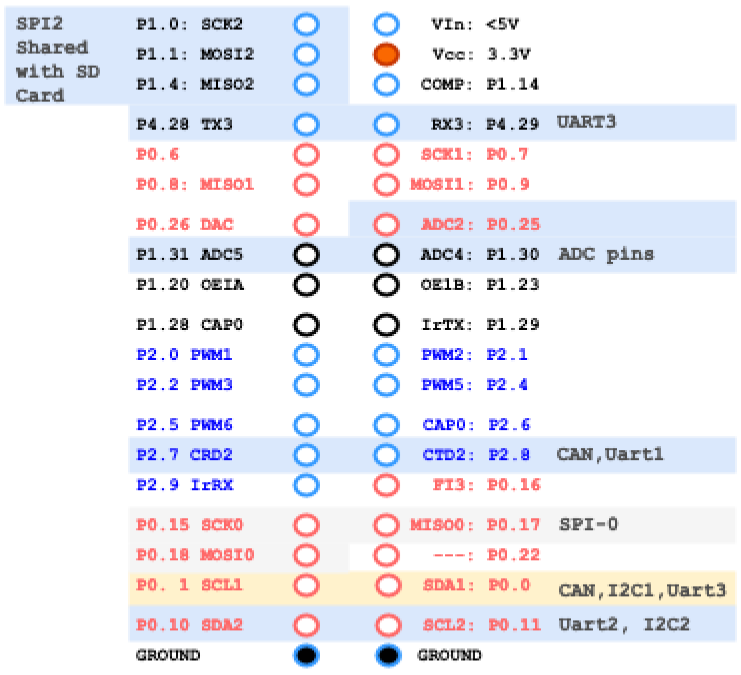


Table 3. Pin description
Not all functions are available on all parts. See Table 2 (Ethernet, USB, LCD, QEI, SD/MMC, comparator pins) and Table 5 (EMC pins).

Symbol	Pin LQFP208	Ball TFBGA208	Ball TFBGA180	Pin LQFP144	Pin LQFP100	Pin LQFP80	Pin TFBGA80		Reset state[1]	Type[2]	Description
P0[0] to P0[31]										I/O	**Port 0:** Port 0 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 0 pins depends upon the pin function selected via the pin connect block.
P0[0]	94	U15	M10	66	46	37	J9	[3]	I; PU	I/O	**P0[0]** — General purpose digital input/output pin.
										I	**CAN_RD1** — CAN1 receiver input.
										O	**U3_TXD** — Transmitter output for UART3.
										I/O	**I2C1_SDA** — I^{2}C1 data input/output (this pin does not use a specialized I2C pad).
										O	**U0_TXD** — Transmitter output for UART0.
P0[1]	96	T14	N11	67	47	38	J10	[3]	I; PU	I/O	**P0[1]** — General purpose digital input/output pin.
										O	**CAN_TD1** — CAN1 transmitter output.
										I	**U3_RXD** — Receiver input for UART3.
										I/O	**I2C1_SCL** — I^{2}C1 clock input/output (this pin does not use a specialized I2C pad).
										I	**U0_RXD** — Receiver input for UART0.

Figure 5: Processor Datasheet pin descriptions for all physical pin connections

This matches the exposed breakout pin P0.0 on our SJ-Two board for the CAN1 peripheral:



Revision #1
Created 2 years ago by isa_team
Updated 2 years ago by isa_team