Raspberry Pi Pico prototype

The aim is to develop a well working prototype based on Raspberry Pi Pico (* 2).

It must wire up,

- RCWL-0517 radar body sensor (https://dronebotworkshop.com/rcwl-0516-experiments/)
- VM3011 MEMS microphone
- Ambient light sensor
- SCCB parallel camera such as OV2710
- NFC antenna via I2C
- Battery voltage measuring
- Headers for measuring power consumption

The software should idle with low power when nothing happens and wake up to respond to sensor input. The implementation should create an event object on IRQ and pass on to a configurable handler chain.

Parts of the prototype

- Pico programmer based on Picoprobe, How-to.
- T-USB Pico that connects to sensors
- Software source code as Git repository
- A physical setup with two Picos
- Instructions/Notes on how to run it

Connect Camera

Connect to an image sensor using the least amount of pins in addition to SCCB(I2C) interface. The purpose is to be able to capture low resolutions still images for simple motion detection using

See PDF

DVP parallel camera interfacing over SPI or SCCB imaging

https://www.arducam.com/raspberry-pi-pico-tensorflow-lite-micro-person-detection-arducam/

https://maker.pro/arduino/tutorial/how-to-interface-the-ov7670-camera-module-with-arduino

https://www.tinyml.org/event/summit-2021/

https://www.tinyml.org/event/summit-2022/

Pico Pin allocation

Pin	Code	ZiBus	T-USB	Probe	Description	Function	SoM Dev Board

Pin	Code	ZiBus	T-USB	Probe	Description	Function	SoM Dev Board
2	GP00				UART_RP_TXD for debugging	F2 UART0	-
3	GP01				UART_RP_RXD for debugging	F2 UART0	-
4	GP02	x			STEM INT Sensor / Expander IRQ		
5	GP03	x			NIGHT INT Sensor IRQ		
6	GP04	x			SYS I2C SDA	F3 I2C0	P1.97
7	GP05	x			SYS I2C SCL	F3 I2C0	P1.99
8	GP06		VM3011 CLK	Probe SWCLK			
9	GP07		VM3011 DATA	Probe SWDIO			
10	GP08		UART1 Tx	UART2 Tx		F2 UART1	P20.1
11	GP09		UART1 Rx	UART2 Rx		F2 UART1	P20.3
12	GP10	x			SDIO CLK / SPI_CLK	F1 SPI1	
13	GP11	x			SDIO CMD / SPI_MOSI or TX	F1 SPI1	
14	GP12		UART3 Tx	UART4 Tx		F2 UART0	P20.2 / P1.61
15	GP13		UART3 Rx	UART4 Rx		F2 UART0	P20.4 / P1.21
16	GP14	x			SDIO DATA 1		
17	GP15	x			SDIO DATA 2		
18	GP16	x			STEM SDA	F3 I2C0	-
19	GP17	x			STEM SCL	F3 I2C0	-
20	GP18	x			Night I2C6 SDA	F3 I2C1	P21.4
21	GP19	x			Night I2C6 SCL	F3 I2C1	P21.2
22	GP20				- Reserved (Clock)	F8 GPIN0	P20.8

Pin	Code	ZiBus	T-USB	Probe	Description	Function	SoM Dev Board
23	GP21				- Reserved (Clock)	F8 GPOUT0	P20.10
24	GP22	x			SDIO DATA 3	F8 CLOCK GPIN1	
25	GP23				- Reserved for Speaker I2S SDO (Dev LED)	F8 CLOCK GPOUT1	
26	GP24			Probe UART TX	- Reserved for Speaker I2S BCK		
27	GP25			Probe UART RX	- Reserved for Speaker I2S LRCK		
28	GP26		x		Reserved Human Body analog signal	F1 SPI1	
29	GP27		x		Reserved Voltage / Analog Sensor	F1 SPI1	
30	GP28	x			SDIO DATA0 / SPI_MISO or RX	F1 SPI1	
31	GP29				SPI_CS, reserved, do not connect	F1 SPI1	
46	USB_DM				Stem MCU 2.0 D-		
47	USB_DP				Stem MCU 2.0 D+		
48	USB_VDD				Not used		

SWD to GP6, GP7, GP29 or 23 - 25 ?

TODO

Human Body monitor Analog pin

RP2040 should probably be SPI master, is MOSI TX or RX ?

Allocate pins for I2S (with clock?)

Debug LED pins?

Communicating between two RP2040s

Two Picos should be wired up with common pins to test SPI/I2C/CAN/QSPI/SDIO data exchange.

SDIO GPIO header pins 13, 15, 16, 18, 22, 37 (suggested) SDIO UCM i.MX8 pins P2 92, 94, 96, 97, 98, 99, 100

To interface with SoM should multiple pins be connected (SPI + SDIO). I.E. P2.89 to P2.97 ?

Twisted MOSI - MISO

- Protocol Info
- Raspberry Pi SDIO connections
- SPI Accessing the SD Card
- Secure Digital (SD) pinouts
-
-

Lines on the bus,

- SDIO CD / SPI CS (only on SD card)
- SDIO CMD / SPI MOSI (.13 SPI1 TX) - verfy?
- SDIO CLK / SPI CLK (.12 SPI1 SCK)
- SDIO DATA0 / SPI MISO (.30 SPI1 RX) - verify?
- SDIO DATA1 (.16 SPI0 RX)
- SDIO DATA2 (.17 SPI1 TX)
- SDIO DATA3 (.24 SPI0 SCK)
- STEM INT (.4)
- STEM SCL (.19)
- STEM SDA (.18)
- NIGHT INT (.5)
- NIGHT SCL (.21)
- NIGHT SDA (.20)
- SYS SCL (.7)
- SYS SDA (.6)

SD Card must ignore other pins if Card Select(CS) isn't enabled. All MCUs must monitor the Card Select(CS) to see if other MCU is running SDIO mode.

- One MCU locks bus in SDIO and becomes master
-

MCU cluster

(?) Structure of SoM with multiple low power MCUs around

SYS I2C bus

The SYS bus is normally managed by the i.MX SoM, so the RPs will normally do nothing with it. However they have pins allocated to allow taking over

Stem I2C bus

The Stem I2C is used to identify connected boards and coordinate the bus setup. Each board will have an EEPROM with unique keys or emulate one with the MCU. Each one with a different address. MCUs on Stem I2C must act as slaves in the address range 0x70 - 0x7f (or 10 bit space?)

To make changes an MCU must write to all other MCUs in the range the address of the MCU taking charge. When done it must write 0 to the same MCUs. When booting an MCU must listen for activity on the data pin for 3 sec before trying to take charge. MCU in charge must use the bus at least once every 2.5 sec. Should there be a time limit to being in charge?

The Stem I2C is accessable on the SoM as I2C3.

An MCU can take charge of,

- SDIO access to SD Card / MMC
- Night I2C
- Monitoring Sensors on the Stem I2C

Chip enable via I/O Expander

MCU I2C slaves on the Stem

Information readable

- Time since startup (16 bit)
- Which address is master (16 bit)
- Product Key

Test SDIO against

- SD Card slot
- MMC
- BT/WiFi modules
- WiFi 5 m.2 Key B
- WiFi 6 m.2 Key E
-