Overview

File: overview.kicad_sch

Mainboard

DEMUX_A0
DEMUX_A1
DEMUX_A2
DEMUX_IE
SSI_SCLK
SSI_SD0
SSI_SD1
SSI_SD2
SSI_SD3

File: board.kicad_sch

PSRAM Array

A0
A1
A2
IE
SCK
SIO0
SIO1
SIO2
SIO3

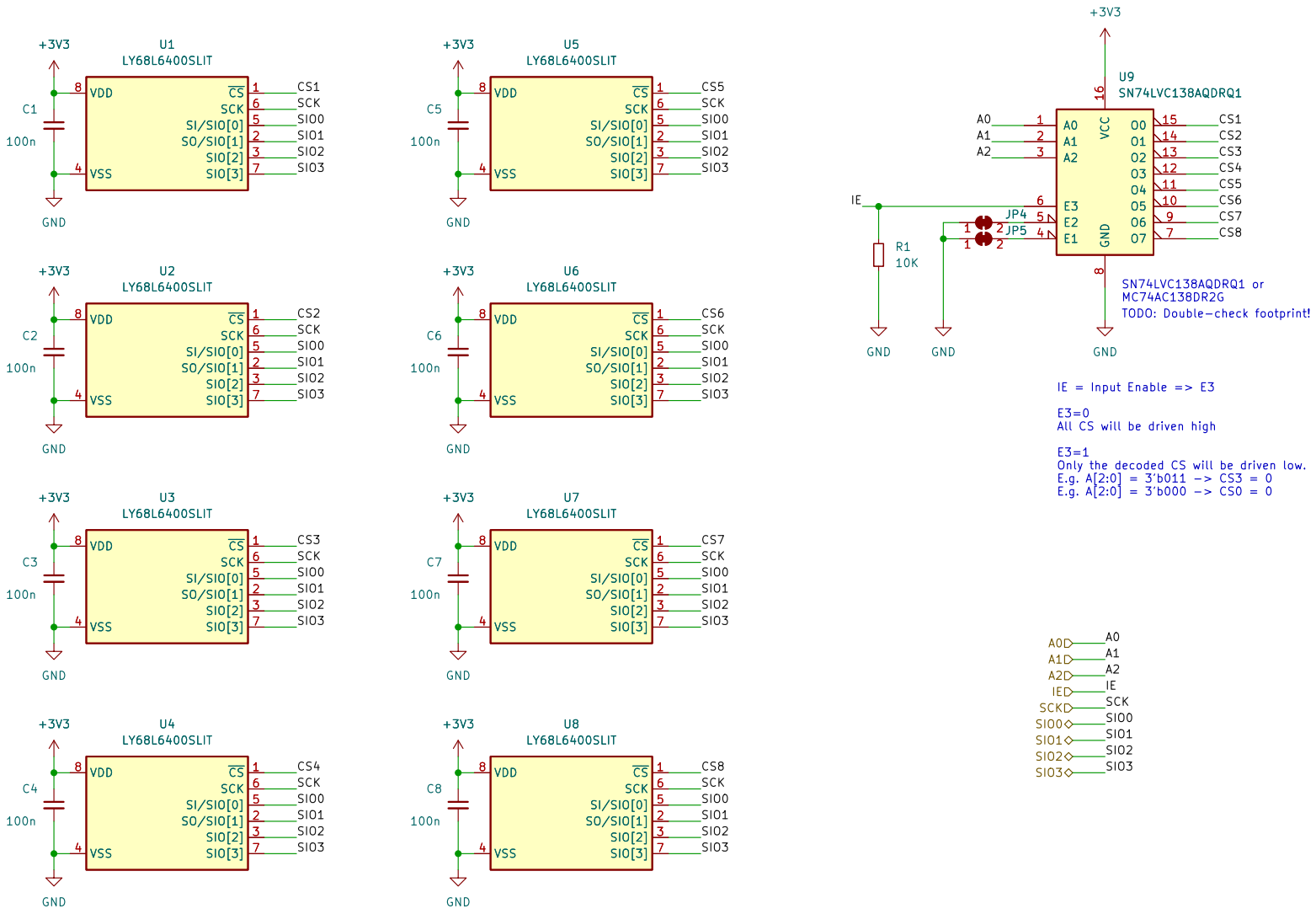
File: psram.kicad_sch

Konrad Beckmann

Sheet: /
File: picocart64_v2.kicad_sch

Title: PicoCart64

Size: A4	Date: 2022-08-14	**Rev: V2.0**
KiCad E.D.A. kicad (6.0.7)		Id: 1/4



Sheet: /PSRAM Array/
File: psram.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. kicad (6.0.7)		Id: 2/4

MCU2 USB connector
J1
USB_C_Receptacle_USB2.0
Ferrite
FB1
VBUS
PWR_FLAG
Compatible with many LDOs
U11
AZ1117-3.3
Q1
FDN340P
N64_VCC
+3V3
C11
10u
C38
10u
3V3_USB
C10
10u
R20
100k
R5
5k1
R6
5k1
MCU2_USB_D-
MCU2_USB_D+
FDN340P is rated for 2A
Prevent N64_VCC to feed into LDO
Prevent 3V3_USB to feed into N64_VCC
If both are connected, pick 3V3_USB
Optional ESD protection
USBLC6-2SC6
U10
R2
100k
DNP - For extra debug/evaluation only
CC resistors omitted for simplicity
J2
USB_C_Receptacle_USB2.0
MCU1_USB_D-
MCU1_USB_D+
R3
100k
J3
Conn_02x06_Odd_Even
MCU2_UART_RX
MCU2_UART_TX
MCU2_SWDIO
MCU2_SWCLK
MCU2_RUN
MCU2_GPIO21
MCU2_CS
MCU1_SWDIO
MCU1_SWCLK
MCU1_RUN
[MCU1_UART_TX]
R7
10K
U13
W25Q128JVS
SSI_CS
SSI_SCLK
SSI_SD0
SSI_SD1
SSI_SD2
SSI_SD3
C39
100n
NOTE: May switch to smaller flash later.
Let's go for 16MB, makes it easier to develop.
D1
WS2812B-2020
MCU1
Handles the N64_BUS
U14
RP2040
C14 100n
C15 100n
C16 100n
C17 100n
C18 100n
C19 100n
C20 1u
C21 100n
C22 100n
C23 1u
NOTE: MCU2 will drive pin high
MCU1_RUN
R9 10K
MCU1_USB_D-
MCU1_USB_D+
R10 22R
R11 22R
MCU1_SWCLK
MCU1_SWDIO
JP1
MCU2_GPIO21
BOM Optimization
MCU2 CLK out replaces Xtal
C12 12p
C13 12p
Y1
12MHz 10ppm
MCU1_XIN
MCU1_XOUT
R12 10K
+3V3 [PIO#2]
TP1
TP2 [MCU1_UART0_TX -> MCU2_RX]
TP3 [MCU1_UART0_RX -> MCU2_TX]
ERC Note:
SSI_SCLK is driven from multuple sources and it's OK.
DEMUX_A0
DEMUX_A1
DEMUX_A2
DEMUX_IE [Pulled Down]
MCU1 <-> MCU2 communication
1. Backup solution, use SPI (PIO on MCU1, SPI slave on MCU2)
2. Backup solution, use UART
MCU2_CS might function as MCU1_INT.
MCU2
Handles ESP32, SDCard
U15
RP2040
SW1
BootButton
Strong pull
R17 1K
MCU2_UART_TX [UART0_TX]
MCU2_UART_RX [UART0_RX]
SD_CLK [SPIO_SCK] [PIO#1]
SD_CMD [SPIO_TX/MOSI/PIO0_IO0]
SD_DAT0 [SPIO_RX/MISO/PIO0_IO1]
SD_DAT1 [PIO0_IO2]
SD_DAT2 [PIO0_IO3]
SD_DAT3 [gpio_CSn/PIO0_IO4]
Either use SPIO with custom CS pin, or do QSPI with PIO
ESP32_D0_TX UART1_TX
ESP32_D1_RX UART1_RX
ESP32_D2_G9
ESP32_D3
ESP32_CS
ESP32_SCK
ESP32_EN
DEMUX_A0
DEMUX_A1
DEMUX_A2
DEMUX_IE
MCU1_RUN [Pulled Down]
LED
MCU2_GPIO21 [clk_gpout0]
COLD_RESET
NMI
CIC_DIO
CIC_DCLK
MCU2_SCK [SPI1_SCK]
MCU2_G27 R16 470R [SPI1_TX]
MCU2_DIO [SPI1_RX] [UART0_TX]
MCU2_CS [Pulled Up] [SPI1_CSn] [UART0_RX]
R18 10k
R13 10k
R14 22R
R15 22R
C26 12p
C27 12p
Y2
12MHz 10ppm
MCU1/MCU2 boot sequence
GPIO26 used to identify MCU1/MCU2
MCU1: Low (pulled down by DEMUX_IE)
MCU2: High (pulled high by MCU2_SCK)
U12
N64-Cartridge
RN1 22R_x4
RN2 22R_x4
RN3 22R_x4
RN4 22R_x4
RN5 22R_x4
RN6 22R_x4
R4 22R
R8 22R
C24 1u
C25 100n
J4
Micro_SD_Card_Det
Optional ESD protection
PUSB3TB6
Project Area
J6
Conn_01x06
J5
Conn_01x02
U16
ESP32-C3-WROOM-02
Pulled Down, powered down by default
R19 10k
EN/CHIP_PU
GPIO0/ADC1_CH0/XTAL_32K_P
GPIO1/ADC1_CH1/XTAL_32K_N
GPIO2/ADC1_CH2
GPIO3/ADC1_CH3
GPIO4/ADC1_CH4
GPIO5/ADC2_CH0
GPIO6
GPIO7
GPIO8
GPIO9
GPIO10
GPIO20/U0RXD
GPIO21/U0TXD
GPIO18/USB_D-
GPIO19/USB_D+
JP2 ESP32_D0_TX
JP3 ESP32_D1_RX
3V3

Sheet: /Mainboard/
File: board.kicad_sch
Title:
Size: A3
Date:
Rev:
KiCad E.D.A. kicad (6.0.7)
Id: 3/4

--- PSRAM Architecture ---
MCU1, MCU2 and all PSRAMs are connected to SSI. Both can read/write.
MCU1 and MCU2 are connected to a 3-bit, 8 output demux + IE pin.
Demux is connected to each PSRAM's CS pin.

--- N64 ROM ---
MCU2 holds MCU1 in reset, fills PSRAM with data.
MCU1 handles N64 bus requests, reads data directly from PSRAM.

--- N64 SRAM ---
MCU1 handles bus request, read/store in internal SRAM.
Stores to external flash directly, and/or to SDCard via MCU2.

--- N64 Serial Interface (EEPROM/RTC) ---
MCU1 handles this on the core that doesn't handle the N64_PI bus.
Stores to external flash directly, and/or to SDCard via MCU2.

--- CIC Emulation ---
MCU2 handles this on a dedicated core.

--- MCU1 -> MCU2 communication ---
MCU2_CS should be configured as input on MCU2, and Open Drain output on MCU1.
MCU1 acts as an SPI Master with PIO.
Combining miso/mosi to one pin, we save 1 pin.

--- Firmware delivery ---
* Small Flash on shared SSI bus
Bootloader firmware for both MCU1 and MCU2 is stored on this

* MCU2:
MCU2 is "main" and will boot first (MCU1_RUN has external pulldown).
After boot, MCU2 realizes it's MCU2 by checking MCU1_RUN pin (will be low)
MCU2 may download application firmware from SDCard into RAM and execute.
MCU2 disables output on SSI_CS and SSI_SCLK.
MCU2 drives MCU1_RUN high, letting MCU1 boot.

* MCU1:
After loading bootloader from flash, detects it's MCU1.
  ! The GPIO chosen for MCU2's [MCU1_RUN GPIO] must be connected to an
    idle-high signal on MCU1, e.g. MCU2's MCU2_CS? !
MCU1 optionally requests application from MCU2 over DIO-SPI.

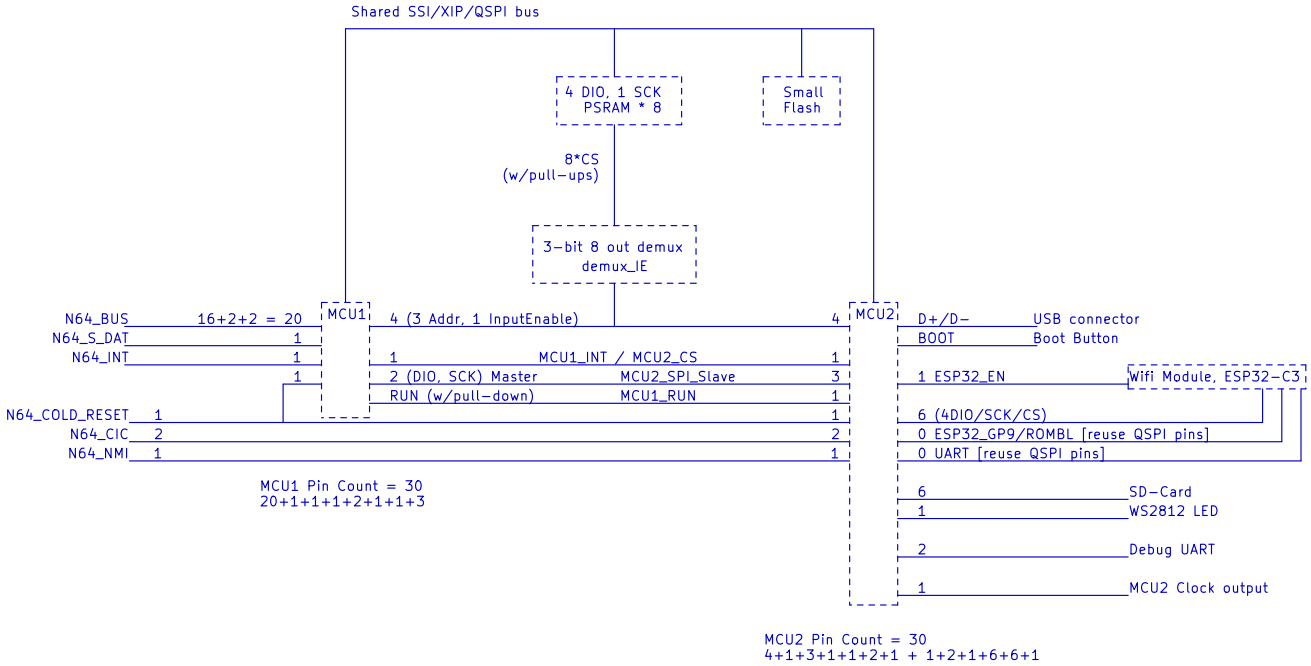
* WiFi module:
Firmware can be updated from MCU2 with a recovery image.

--- WiFi module role ---
MCU2 can update firmware by pulling GP9 low, toggling EN,
then perform update over UART.
WiFi module enables loading ROMs from network to SDCard.
WiFi module enables saveing/loading SRAM/EEPROM saves to network.
Potential multiplayer stuff.
MCU1 can access it via MCU2.
MCU2 uses PIO QSPI controller.
QSPI slave interface can be implemented and assigned on any GPIO.

---------- Core responsibilities

MCU1:
- Core 0:
--- N64_BUS [PIO #1]
--- N64_INT control
--- N64_CR interrupt
--- PSRAM synchronous access
- Core 1:
--- Asynchronous tasks
--- MCU2 SPI [PIO #2]
--- N64_S_DAT communication

MCU2:
- Core 0:
--- PSRAM
--- SDCard [PIO #1]
--- ESP32 QSPI [PIO #2]
--- LED control
--- N64_NMI control
--- SPI slave (MCU1 is master)
- Core 1:
--- CIC
--- N64_CR interrupt



MCU1 Pin Count = 30
20+1+1+1+2+1+1+3

MCU2 Pin Count = 30
4+1+3+1+1+2+1 + 1+2+1+6+6+1

MC74AC138 fast 3-bit 8 output demultiplexer (active LOW)

Alternative demux
sn74lvc138a-q1

PSRAM alternative to ly68l6400slit:
APS6404L-3SQR-SN

RP2040 internal SRAM = 264 kB

Sheet: /Overview/		
File: overview.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. kicad (6.0.7)		Id: 4/4