

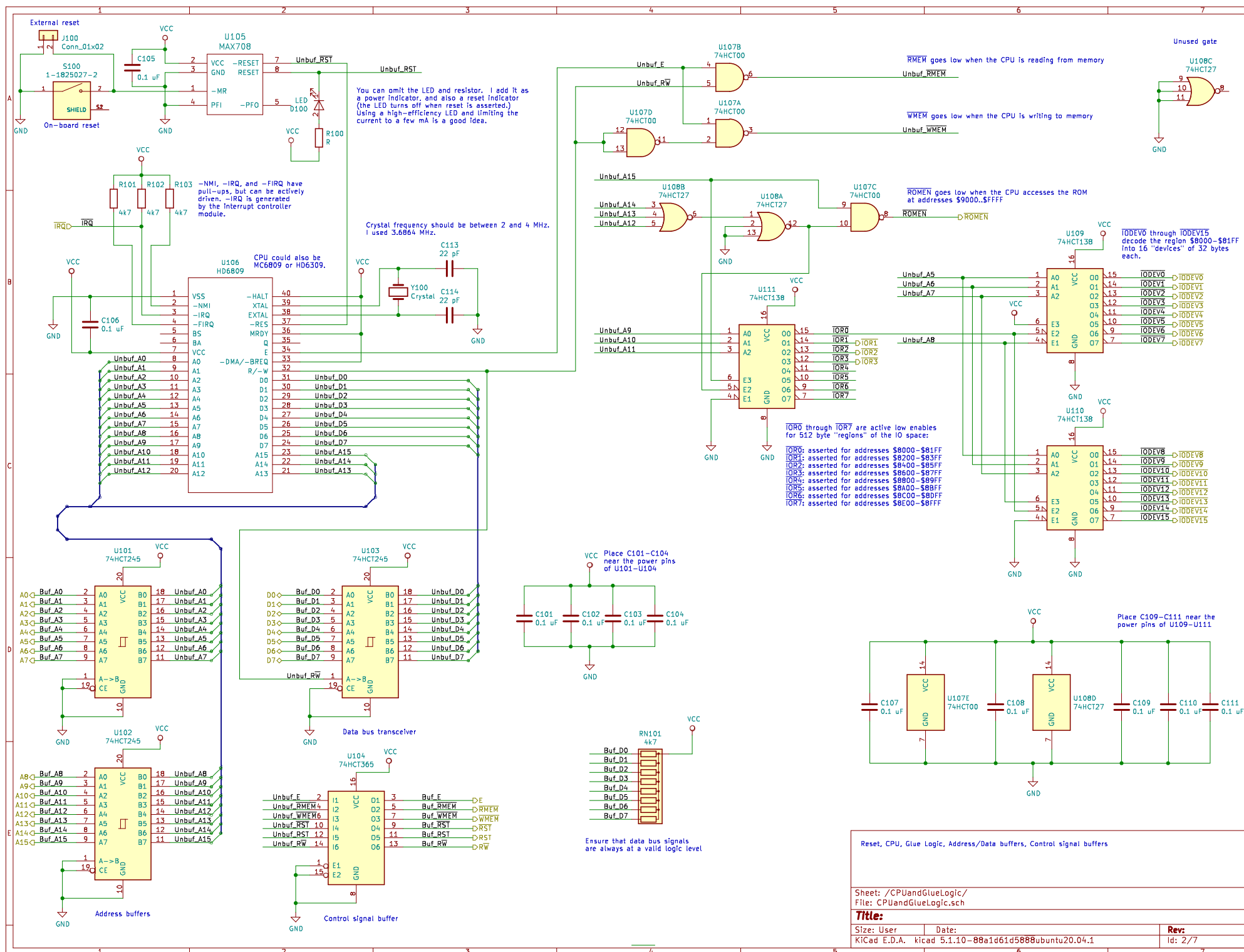
F68: A DIY 6809 microcomputer system

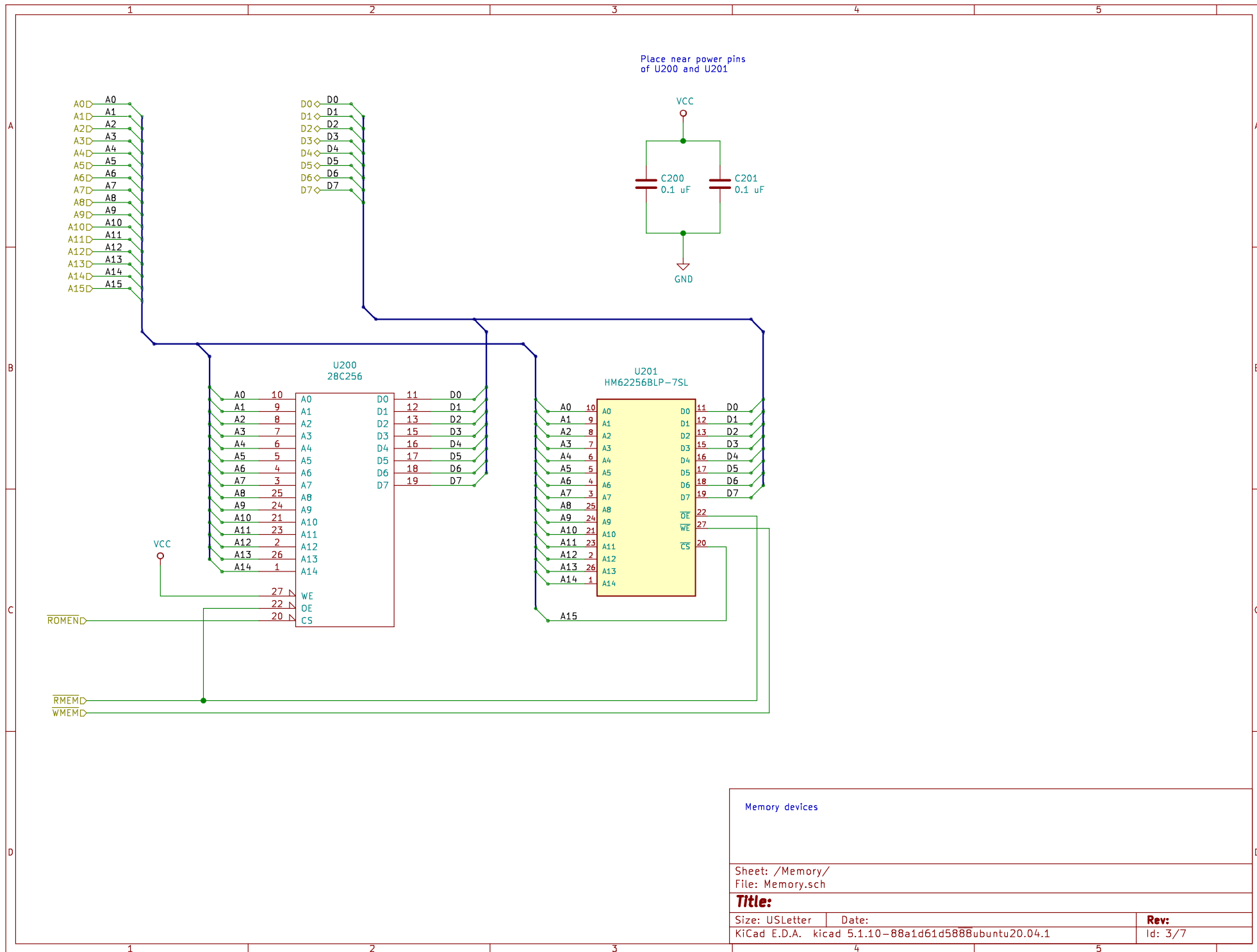
Sheet: /
File: F68.sch

Title:

Size: User Date: KiCad E.D.A. kicad 5.1.10-88a1d61d5888ubuntu20.04.1

Rev: Id: 1/7





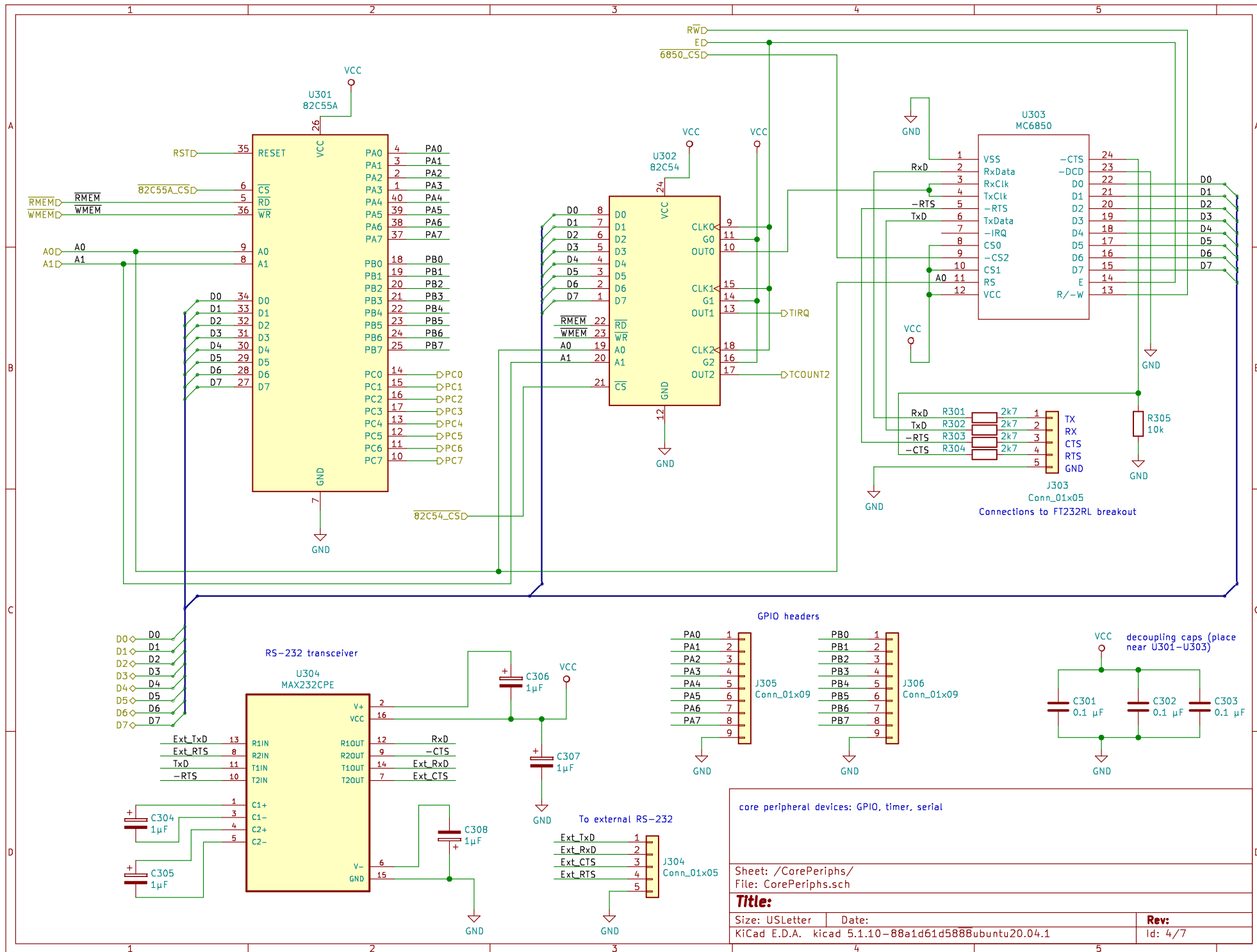
Memory devices

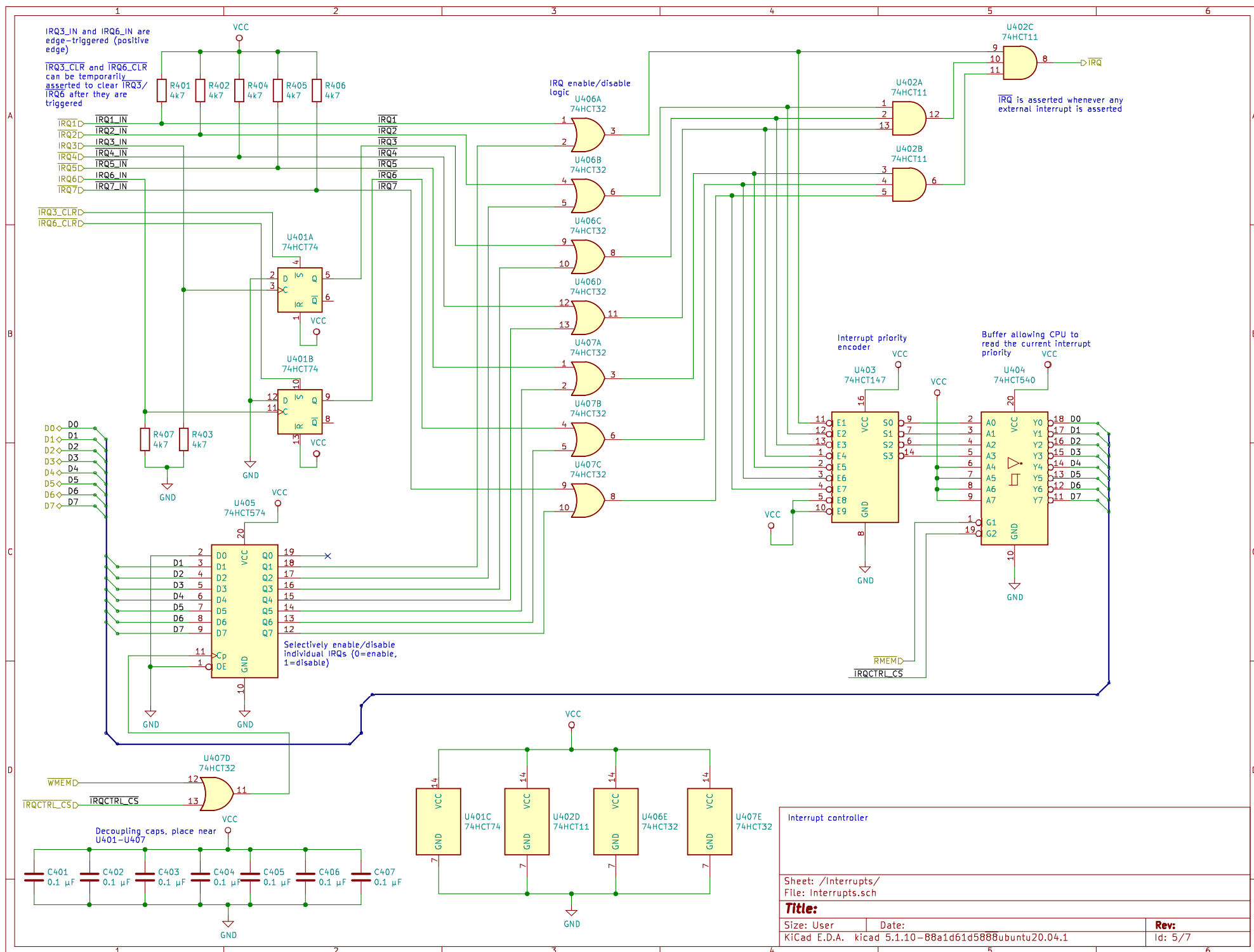
Sheet: /Memory/
File: Memory.sch

Title:

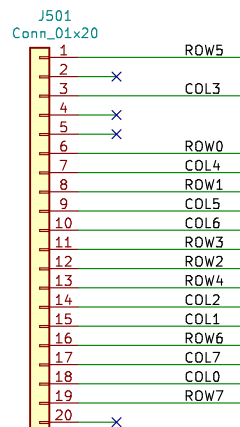
Size: USLetter Date:
KiCad E.D.A. kicad 5.1.10-88a1d61d5888ubuntu20.04.1

Rev:
Id: 3/7

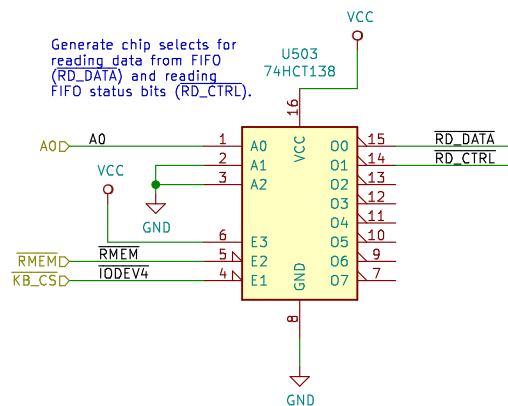




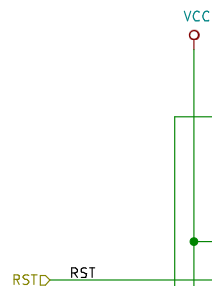
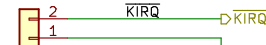
C16 keyboard connector, pin 2 is a key and should be removed



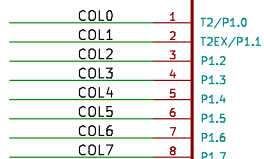
Generate chip selects for reading data from FIFO (RD_DATA) and reading FIFO status bits (RD_CTRL).



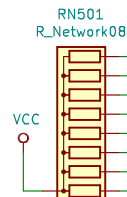
J503 Jumper to enable keyboard IRQ (when FIFO is not empty)



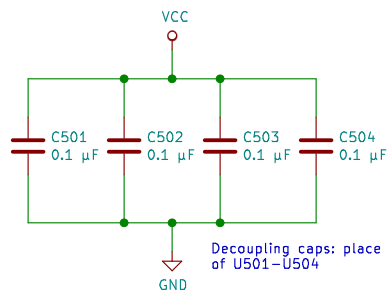
Port 1 actively drives the columns of the keyboard matrix: outputs are normally at VCC, but a column output is driven to GND when a scan of that column is in progress



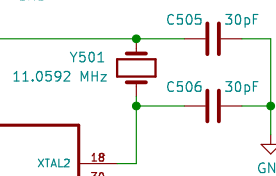
Port 0 reads the rows (with pull up resistors keeping the row inputs at VCC unless a keyswitch has made a connection to a column when the column scan is in progress)



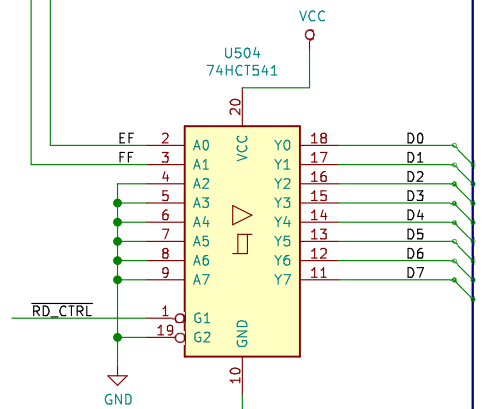
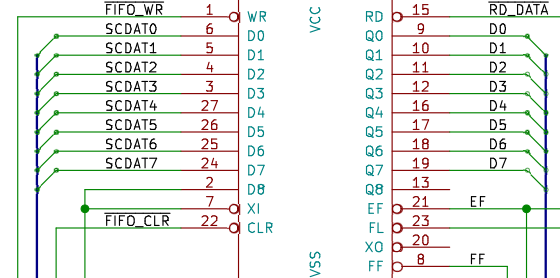
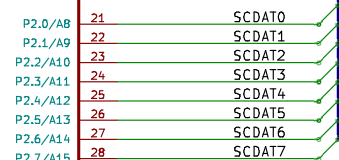
The exact value of RN501 isn't critical: anything between 2k7 and 47k should work fine.



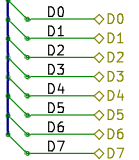
Microcontroller to scan keyboard matrix and send scan codes to FIFO. Will be a 89C51 or 87C51.



Port 2 is used for sending scan code data to the FIFO chip.



Buffer to allow EF (empty flag) and FF (full flag) bits of FIFO to be read by the CPU.



daveho hacks

Sheet: /Keyboard/
File: Keyboard.sch

Title: keyboard controller

Size: USLetter Date:
KiCad E.D.A. kicad 5.1.10-88a1d61d588bubuntu20.04.1

Rev:
Id: 6/7

