Module 1: core (CPU and glue logic)

Sheet: CPUandGlueLogic

File: CPUandGlueLogic.sch

Module 2: memory and core peripherals

Sheet: Memory

File: Memory.sch

Sheet: CorePeriphs

File: CorePeriphs.sch

Module 3: interrupt controller

Sheet: Interrupts

File: Interrupts.sch

Module 4: keyboard controller

Sheet: Keyboard

File: Keyboard.sch

Sheet: Display

File: Display.sch

Backplane connector: one per board PCB
(appears here because all backplane signals
exist on this sheet)

back row (backplane)
bottom row (board)

front row (backplane)
top row (board)

J001
Conn_02x40_Odd_Even

		1	2		
	IRQ	3	4		D0
	PC0	5	6		D1
	PC1	7	8		D2
	PC2	9	10		D3
	PC3	11	12		D4
	PC4	13	14		D5
	PC5	15	16		D6
	PC6	17	18		D7
	PC7	19	20		A8
	IRQ3	21	22		A9
	IRQ1	23	24		A10
	IRQ2	25	26		A11
	IRQ4	27	28		A12
	IRQ5	29	30		A13
	IRQ6	31	32		A14
	IRQ7	33	34		A15
	TCOUNT2	35	36		A0
		37	38		A1
		39	40		A2
	IOR1	41	42		A3
	IOR2	43	44		A4
	IOR3	45	46		A5
	IODEV0	47	48		A6
	IODEV1	49	50		A7
	IODEV2	51	52		
	IODEV3	53	54		
	IODEV4	55	56		
	IODEV5	57	58		
	IODEV6	59	60		
	IODEV7	61	62		
	IODEV8	63	64		RAMEN
	IODEV9	65	66		RW
	IODEV10	67	68		ROMEN
	IODEV11	69	70		E
	IODEV12	71	72		RMEM
	IODEV13	73	74		WMEM
	IODEV14	75	76		RST
	IODEV15	77	78		RST
		79	80		

VCC
GND
VCC
C001
10µF
GND

Bulk decoupling for board,
place near power pins of
2x40 connector to backplane

F68: A DIY 6809 microcomputer system

Sheet: /
File: F68.sch

Title:

Size: User | Date: | Rev:

KiCad E.D.A. kicad 5.1.10-88a1d61d5890ubuntu20.04.1 | Id: 1/7

External reset

J100
Conn_01x02

S100
1-1825027-2
SHIELD
On-board reset

C105
0.1 uF

U105
MAX708
VCC
GND
-RESET
RESET
-MR
PFI
-PFO

Unbuf_RST

LED
D100

R100
R

You can omit the LED and resistor. I add it as a power indicator, and also a reset indicator (the LED turns off when reset is asserted.) Using a high-efficiency LED and limiting the current to a few mA is a good idea.

R101 R102 R103
4k7 4k7 4k7

-NMI, -IRQ, and -FIRQ have pull-ups, but can be actively driven. -IRQ is generated by the interrupt controller module.

IRQ

Crystal frequency should be between 2 and 4 MHz. I used 3.6864 MHz.

U106
HD6809

CPU could also be MC6809 or HD6309.

Y100
Crystal

C113
22 pF

C114
22 pF

C106
0.1 uF

VSS
-NMI
-IRQ
-FIRQ
BS
BA
VCC
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
-HALT
XTAL
EXTAL
-RES
MRDY
Q
E
-DMA/-BREQ
R/-W
D0
D1
D2
D3
D4
D5
D6
D7
A15
A14
A13

Unbuf_A0 Unbuf_A1 Unbuf_A2 Unbuf_A3 Unbuf_A4 Unbuf_A5 Unbuf_A6 Unbuf_A7 Unbuf_A8 Unbuf_A9 Unbuf_A10 Unbuf_A11 Unbuf_A12

Unbuf_D0 Unbuf_D1 Unbuf_D2 Unbuf_D3 Unbuf_D4 Unbuf_D5 Unbuf_D6 Unbuf_D7 Unbuf_A15 Unbuf_A14 Unbuf_A13

U101
74HCT245

U102
74HCT245

Address buffers

U103
74HCT245

Buf_A0 Buf_A1 Buf_A2 Buf_A3 Buf_A4 Buf_A5 Buf_A6 Buf_A7 Buf_A8 Buf_A9 Buf_A10 Buf_A11 Buf_A12 Buf_A13 Buf_A14 Buf_A15

Unbuf_A0 Unbuf_A1 Unbuf_A2 Unbuf_A3 Unbuf_A4 Unbuf_A5 Unbuf_A6 Unbuf_A7 Unbuf_A8 Unbuf_A9 Unbuf_A10 Unbuf_A11 Unbuf_A12 Unbuf_A13 Unbuf_A14 Unbuf_A15

Buf_D0 Buf_D1 Buf_D2 Buf_D3 Buf_D4 Buf_D5 Buf_D6 Buf_D7

Unbuf_RW

Data bus transceiver

U104
74HCT365

Unbuf_E
Unbuf_RMEM
Unbuf_WMEM
Unbuf_RST
Unbuf_RST
Unbuf_RW

Buf_E
Buf_RMEM
Buf_WMEM
Buf_RST
Buf_RST
Buf_RW

E
RMEM
WMEM
RST
RST
RW

Control signal buffer

U107B
74HCT00

U107A
74HCT00

U107D
74HCT00

Unbuf_E
Unbuf_RW

Unbuf_RMEM
Unbuf_WMEM

RMEM goes low when the CPU is reading from memory

WMEM goes low when the CPU is writing to memory

Unused gate

U108C
74HCT27

U108B
74HCT27

U108A
74HCT27

U107C
74HCT00

Unbuf_A15
Unbuf_A14
Unbuf_A13
Unbuf_A12

ROMEN

ROMEN goes low when the CPU accesses the ROM at addresses $9000..$FFFF

U111
74HCT138

Unbuf_A9
Unbuf_A10
Unbuf_A11

IOR0 IOR1 IOR2 IOR3 IOR4 IOR5 IOR6 IOR7

IOR0 through IOR7 are active low enables for 512 byte "regions" of the IO space:

IOR0: asserted for addresses $8000-$81FF
IOR1: asserted for addresses $8200-$83FF
IOR2: asserted for addresses $8400-$85FF
IOR3: asserted for addresses $8600-$87FF
IOR4: asserted for addresses $8800-$89FF
IOR5: asserted for addresses $8A00-$8BFF
IOR6: asserted for addresses $8C00-$8DFF
IOR7: asserted for addresses $8E00-$8FFF

U109
74HCT138

Unbuf_A5
Unbuf_A6
Unbuf_A7
Unbuf_A8

IODEV0 IODEV1 IODEV2 IODEV3 IODEV4 IODEV5 IODEV6 IODEV7

IODEV0 through IODEV15 decode the region $8000-$81FF into 16 "devices" of 32 bytes each.

U110
74HCT138

IODEV8 IODEV9 IODEV10 IODEV11 IODEV12 IODEV13 IODEV14 IODEV15

Place C101-C104 near the power pins of U101-U104

C101 C102 C103 C104
0.1 uF

RN101
4k7

Buf_D0 Buf_D1 Buf_D2 Buf_D3 Buf_D4 Buf_D5 Buf_D6 Buf_D7

Ensure that data bus signals are always at a valid logic level

Place C109-C111 near the power pins of U109-U111

C107 C108 C109 C110 C111
0.1 uF

U107E
74HCT00

U108D
74HCT27

Reset, CPU, Glue Logic, Address/Data buffers, Control signal buffers

Sheet: /CPUandGlueLogic/
File: CPUandGlueLogic.sch

Title:

Size: User | Date: | Rev:
KiCad E.D.A. kicad 5.1.10-88a1d61d5890ubuntu20.04.1 | Id: 2/7

Place near power pins
of U200 and U201

VCC

C200
0.1 uF

C201
0.1 uF

GND

A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15

D0 D1 D2 D3 D4 D5 D6 D7

U200
28C256

U201
HM62256BLP-7SL

VCC

ROMEN

RMEM

WMEM

Jumper to select either A15 or $\overline{\text{RAMEN}}$
for enabling the RAM; provides flexibility
for CPU/glue logic modules that have
a different memory map (different than
what the F68 6809 CPU/glue logic module
implements)

J201
Conn_01x03

$\overline{\text{RAMEN}}$

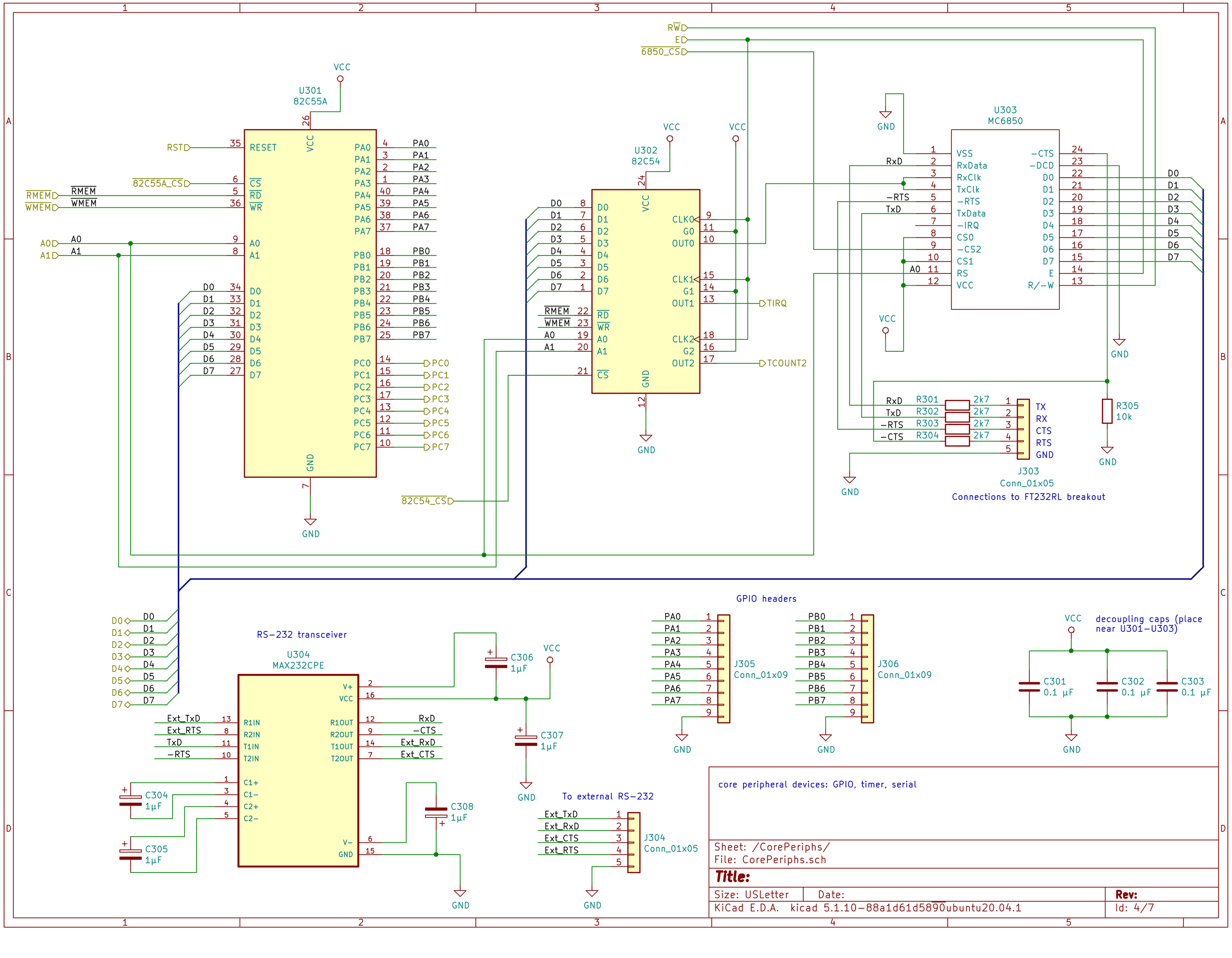
A15

Memory devices

Sheet: /Memory/
File: Memory.sch

Title:

Size: USLetter | Date: | **Rev:**
KiCad E.D.A. kicad 5.1.10-88a1d61d5890ubuntu20.04.1 | Id: 3/7


U301
82C55A
RESET
VCC
GND
RST
82C55A_CS
RMEM
WMEM
A0
A1
D0
D1
D2
D3
D4
D5
D6
D7
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
82C54_CS
U302
82C54
CLK0
G0
OUT0
CLK1
G1
OUT1
CLK2
G2
OUT2
TIRQ
TCOUNT2
RW
E
6850_CS
U303
MC6850
VSS
RxData
RxClk
TxClk
−RTS
TxData
−IRQ
CS0
−CS2
CS1
RS
R/−W
−CTS
−DCD
RxD
TxD
R301
R302
R303
R304
2k7
R305
10k
TX
RX
CTS
RTS
J303
Conn_01x05
Connections to FT232RL breakout
RS−232 transceiver
U304
MAX232CPE
R1IN
R2IN
T1IN
T2IN
R1OUT
R2OUT
T1OUT
T2OUT
V+
V−
C1+
C1−
C2+
C2−
Ext_TxD
Ext_RTS
Ext_RxD
Ext_CTS
C304
C305
C306
C307
C308
1µF
To external RS−232
J304
Conn_01x05
GPIO headers
J305
Conn_01x09
J306
Conn_01x09
decoupling caps (place near U301−U303)
C301
C302
C303
0.1 µF
core peripheral devices: GPIO, timer, serial
Sheet: /CorePeriphs/
File: CorePeriphs.sch
Title:
Size: USLetter
Date:
Rev:
KiCad E.D.A. kicad 5.1.10−88a1d61d5890ubuntu20.04.1
Id: 4/7

IRQ3_IN and IRQ6_IN are edge-triggered (positive edge)

IRQ3_CLR and IRQ6_CLR can be temporarily asserted to clear IRQ3/IRQ6 after they are triggered

R401 4k7
R402 4k7
R404 4k7
R405 4k7
R406 4k7

IRQ1 IRQ1_IN
IRQ2 IRQ2_IN
IRQ3 IRQ3_IN
IRQ4 IRQ4_IN
IRQ5 IRQ5_IN
IRQ6 IRQ6_IN
IRQ7 IRQ7_IN

IRQ3_CLR
IRQ6_CLR

U401A 74HCT74
U401B 74HCT74

R407 4k7
R403 4k7

D0 D1 D2 D3 D4 D5 D6 D7

U405 74HCT574

Selectively enable/disable individual IRQs (0=enable, 1=disable)

IRQ enable/disable logic

U406A 74HCT32
U406B 74HCT32
U406C 74HCT32
U406D 74HCT32
U407A 74HCT32
U407B 74HCT32
U407C 74HCT32

U402C 74HCT11
U402A 74HCT11
U402B 74HCT11

IRQ is asserted whenever any external interrupt is asserted

Interrupt priority encoder

U403 74HCT147

Buffer allowing CPU to read the current interrupt priority

U404 74HCT540

RMEM
IRQCTRL_CS

WMEM
IRQCTRL_CS

U407D 74HCT32

Decoupling caps, place near U401–U407

C401 0.1 µF
C402 0.1 µF
C403 0.1 µF
C404 0.1 µF
C405 0.1 µF
C406 0.1 µF
C407 0.1 µF

U401C 74HCT74
U402D 74HCT11
U406E 74HCT32
U407E 74HCT32

Interrupt controller

Sheet: /Interrupts/
File: Interrupts.sch

Title:

Size: User
Date:
Rev:
KiCad E.D.A. kicad 5.1.10-88a1d61d5890ubuntu20.04.1
Id: 5/7

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

J501
Conn_01x20

1	ROW5	
2		
3	COL3	
4		
5		
6	ROW0	
7	COL4	
8	ROW1	
9	COL5	
10	COL6	
11	ROW3	
12	ROW2	
13	ROW4	
14	COL2	
15	COL1	
16	ROW6	
17	COL7	
18	COL0	
19	ROW7	
20		

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

U503
74HCT138

VCC
GND
A0
RMEM
IODEV4
KB_CS
RD_DATA
RD_CTRL

J503
Conn_01x02

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

KIRQ

U502
IDT7201

FIFO_WR
SCDAT0
SCDAT1
SCDAT2
SCDAT3
SCDAT4
SCDAT5
SCDAT6
SCDAT7
FIFO_CLR
RD_DATA
D0
D1
D2
D3
D4
D5
D6
D7
EF
FF

U501
87C51/89C51

C505 30pF
C506 30pF
Y501
11.0592 MHz

RST
XTAL1
XTAL2
ALE
PSEN
TXD/P3.1
WR/P3.6
RD/P3.7
FIFO_WR
FIFO_CLR
SCDAT0
SCDAT1
SCDAT2
SCDAT3
SCDAT4
SCDAT5
SCDAT6
SCDAT7

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

COL0
COL1
COL2
COL3
COL4
COL5
COL6
COL7
T2/P1.0
T2EX/P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7

RN501
R_Network08

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)

ROW0
ROW1
ROW2
ROW3
ROW4
ROW5
ROW6
ROW7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7

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

RXD/P3.0
INT0/P3.2
INT1/P3.3
T0/P3.4
T1/P3.5
VSS

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

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

U504
74HCT541

EF
FF
RD_CTRL
D0
D1
D2
D3
D4
D5
D6
D7

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

C501 0.1 µF
C502 0.1 µF
C503 0.1 µF
C504 0.1 µF

Decoupling caps: place near power pins
of U501–U504

daveho hacks

Sheet: /Keyboard/
File: Keyboard.sch

Title: keyboard controller

Size: USLetter	Date:	**Rev:**
KiCad E.D.A. kicad 5.1.10-88a1d61d5890ubuntu20.04.1		Id: 6/7

Warning! The signal names for package pins
41 and 42 were reversed on the silkscreen.
Pin 41 is the 12 MHz clock output, and
pin 42 is the third ground pin.

Resistor DAC

U601
Upduino_3.0

VGA breakout

J601
Conn_01x12

TB605
ARDUINO_NANO

The Arduino is just standing
in for the host system for now

U602
IDT7201

U603
74LVC245

U604
74LVC245

Sheet: /Display/	
File: Display.sch	
Title: Display	
Size: User	Date:
KiCad E.D.A. kicad 5.1.10-88a1d61d5890ubuntu20.04.1	Id: 7/7