Revision Updates

BNO
BNO.SchDoc
32kHz_CLK
BNO_I2C
BNO_RST
BNO_INT
13 mA Max @3.3V

STM
STM32F405.SchDoc
BNO_I2C
32kHz_CLK
25MHz_CLK
F4_I/O
SYS_I2C
EC_INT
111 mA Max @3.3V

PSoC
PSoC5LP.SchDoc
F4_I/O
BNO_RST
EC_INT
USB_DnStr
KP_SWD
Atmel_ISP
USB_BRIDGE_nRST
SD_CARD_nPWR_CTRL
Logger_UART_RX_in
Logger_UART_TX_out
BLE_I/O
SYS_I2C
F3_I/O
25 mA Max @3.3V

BLE
BLE.SchDoc
F4_I/O
KP_SWD
32kHz_CLK
25MHz_CLK
TX2
BLE_I/O
SYS_I2C
BNO_I2C
BNO_INT
50 mA Max @3.3V

Serial Data Logger
DataLogger.SchDoc
USB_UpStr
USB_DnStr
Logger_UART_TX_out
Logger_UART_RX_in
SD_CARD_nPWR_CTRL
USB_BRIDGE_nRST
Atmel_ISP
448 mA Max @3.3V

External_Conns
External_Conns.SchDoc
SYS_I2C
F3_I/O
TX2
USB_UpStr
EC_INT
647 mA Max @3.3V
Req'd

MH1 MH2 MH3 MH4 MH5 MH6

Fiducial Mark
Fiducial Mark
Fiducial Mark

TP3 TP5
DGND
3p3V
TP4

Title	***mk8p0 MCU***		
Size: B	Number:1	Revision:REV-2	
Date: 6/17/2020	Time: 3:20:05 PM	Sheet 1 of 7	
File: C:\Temp\Delete\Nimbus\mk8p0_MCU\MCU_Top_Lvl.SchDoc			

Nimbus Robotics
4620 Henry St
Pittsburgh, PA 15213

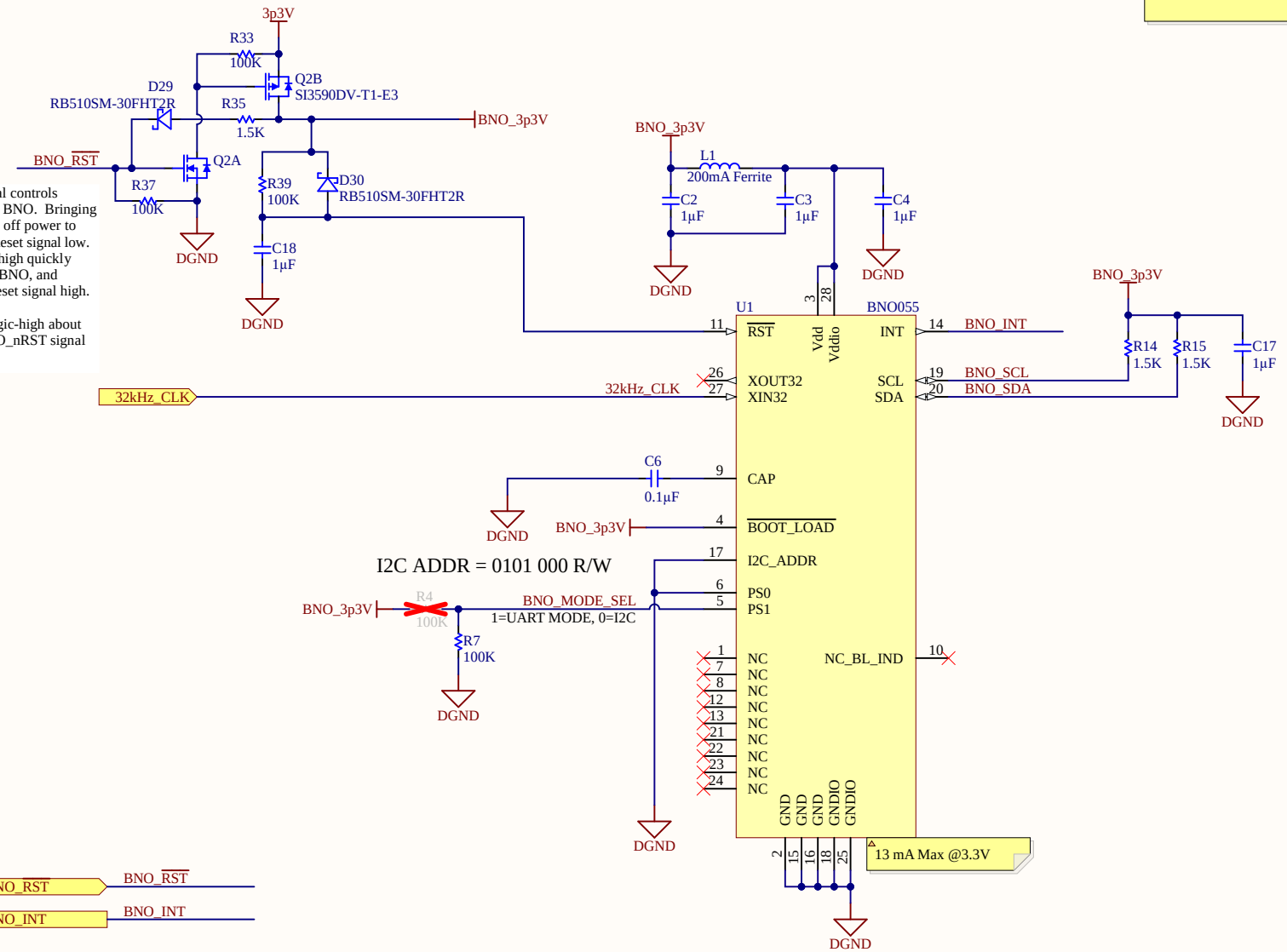
Nimbus Robotics

13 mA Max @3.3V

Revision Updates

The BNO_nRST signal controls power and reset to the BNO. Bringing BNO_nRST low turns off power to BNO and brings the Reset signal low. Bringing BNO-nRST high quickly turns on power to the BNO, and "slowly" brings the Reset signal high.

The Reset pin goes logic-high about 50-100msec after BNO_nRST signal goes high.



Title	***mk8p0 MCU***		
Size: A	Number: 2	Revision: REV-2	
Date: 6/17/2020	Time: 3:20:05 PM	Sheet 2 of 7	
File: C:\Temp\Delete\Nimbus\mk8p0_MCU\BNO.SchDoc			

Nimbus Robotics
4620 Henry St
Pittsburgh, PA 15213

Nimbus Robotics

111 mA Max @3.3V

Revision Updates

UART1: Algo sniffer
UART2: BNO serial option
UART3: BLE and Serial-Bootloading
UART4: MotorCTRL COMMS
UART5: ----------
UART6: Console

I2C1: BNO
I2C2: SYS_I2C
I2C3: FRAM

F4_I/O
F4_nRST
F4_BOOT0
L/R_SELECT
F4_UART1_TX
F4_UART1_RX
F4_UART2_TX
F4_UART2_RX
F4_UART3_TX
F4_UART3_RX
F4_UART4_TX
F4_UART4_RX
F4_UART6_TX
F4_UART6_RX
F4_PB13 — UART3_CTS
1-WIRE
F4_PC1 — TP1
F4_PC2 — BT_MOD_TP3
F4_PC3 — DBG_LED_4
F4_PC4 — F4_BNO_nRST
F4_PC13 — BT_MOD_TP4

BNO_I2C: BNO_SCL, BNO_SDA
SYS_I2C: SYS_SCL, SYS_SDA

Bootloader Pattern: BOOT0=1, BOOT1=0
Normal Operation: BOOT0=0, BOOT1=0

TP2
R20 1.5K
R10 330
D9 3.3V
D1 Blue
C15 2.2µF
C16 2.2µF
C19 0.1µF
DGND

3p3V
C9 0.1µF
C10 0.1µF
C11 0.1µF
C12 0.1µF
C13 0.1µF
C14 0.1µF

U3
STM32F405RGT6TR

RST
BOOT0
PB2/BOOT1
Vcap1
Vcap2
PB0
PB1
PB3/JTDO/TRACESWO
PB4/NJTRST
PB5
PB6
PB7
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
PD2

PA0/WKUP
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13/JTMS-SWDIO
PA14/JTCK-SWCLK
PA15/JTDI

PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
PC10
PC11
PC12
PC13
PC14/OSC32_IN
PC15/OSC32_OUT
PH0/OSC_IN
PH1/OSC_OUT

VBAT
Vdda
Vdd
Vdd
Vdd
Vdd
Vssa
Vss
Vss

109 mA MAX
@168MHz, 105C, all
periphs enabled

F4_nRST
F4_BOOT0
L/R_SELECT
TDO
F4_UART1_TX
F4_UART1_RX
BNO_SCL
BNO_SDA
SYS_SCL
SYS_SDA
EC_INT
R8 1.5K
UART3_CTS
F4_PB13
F4_PB14

F4_UART4_TX
F4_UART4_RX
F4_UART2_TX
F4_UART2_RX
I2C3_SCL
TMS
TCK
F4_PC1
F4_PC2
F4_PC3
F4_PC4
F4_UART6_TX
F4_UART6_RX
I2C3_SDA
F4_UART3_TX
F4_UART3_RX
F4_PC13
32kHz_CLK
25MHz_CLK

R11 1.5K
R13 1.5K
R24 1.5K
R12 1.5K
TP1
BT_MOD_TP3
DBG_LED_4
F4_BNO_nRST
D10 3.3V

U4
IO NC
GND
DS28E05
1 mA Max @3.3V

F4_PC1 — D2 Blue — R21 330
F4_PB14 — D3 Orange — R22 330
F4_PC2 — D4 Green — R27 330
F4_PC3 — D5 Red — R30 330

SWD

3p3V
R23 100K
R29 100K
D12 3.3V
D17 3.3V
D18 3.3V
D19 3.3V
J1
TMS
TCK
TDO
F4_nRST
HEADER 1x6 50mil
DGND

FRAM

3p3V
865-1275-1-ND
U5 MB85RC64TAPN-G
C20 0.1µF
A0
A1
A2
Vss
Vcc
SCL
SDA
WP
I2C3_SCL
I2C3_SDA
ADDR: 1010 000 R/W
1 mA Max @3.3V
DGND

SIGNAL COLOR LEGEND
BLE<-->F4 SIGNALS
BLE<-->PSoC5 SIGNALS
F4<-->PSoC5 SIGNALS

Title	mk8p0 MCU		
Size: B	Number: 3	Revision: REV-2	
Date: 6/17/2020	Time: 3:20:05 PM	Sheet 3 of 7	
File: C:\Temp\Delete\Nimbus\mk8p0_MCU\STM32F405.SchDoc			

Nimbus Robotics
4620 Henry St
Pittsburgh, PA 15213

Nimbus Robotics

25 mA Max @3.3V

BNO_RST — BNO_RST

PSoC5 needs to be able to reset BNO during low-power. Also, PSoC5 will "forward" the reset signals from F4 or M4

Logger_UART_RX_in — Logger_UART_RX_in
Logger_UART_TX_out — Logger_UART_TX_out

PSoC5 will route the appropriate UART lines to the datalogger

USB_BRIDGE_nRST — USB_BRIDGE_nRST
SD_CARD_nPWR_CTRL — SD_CARD_nPWR_CTRL

PSoC5 needs to be able to keep the USB-Bridge in reset during the time when USB is not connected, that's why it controls this Reset line.

Also, the PSoC5 needs to be able to cycle power on the actual card so that the card can switch protocols between SPI/SDIO when needed.

Atmel_ISP — Atmel_RST, Atmel_MOSI, Atmel_MISO, Atmel_SCK

PSoC5 needs to be able to keep the Atmel in reset during the time when USB is connected. It also needs to "forward" the reset signal (and SPI signals) for ISP-programming of the Atmel; these signals will be "forwarded" from the test-point connector of the PSoC5

KP_SWD — KP_SWDIO, KP_SWDCLK, KP_SWDRST

PSoC5 can be configured as a KitProg3 device for programming the BLE.

EC_INT — EC_INT

3p3V
R2 24K
U2 TLV803SDBZT (RESET, VDD, GND)
R3 12K — PSoC5_SWDRST
C8 0.1µF
C7 0.1µF
DGND

U6A CY8C5868LTI-LP039

PSoC5_SWDRST	10	XRES	
Logger_UART_TX_out	48	P0[0]	P1[0]
Atmel_SCK	49	P0[1]	P1[1]
KP_P0_2	50	P0[2]	P1[2]
KP_P0_3	51	P0[3]	P1[3]
KP_P0_4	53	P0[4]	P1[4]
F4_nRST	54	P0[5]	P1[5]
F3_RX1_in	55	P0[6]	P1[6]
F4_PC4	56	P0[7]	P1[7]
Atmel_RST	62	P2[0]	P3[0]
BNO_RST	63	P2[1]	P3[1]
USB_BRIDGE_nRST	64	P2[2]	P3[2]
SD_CARD_nPWR_CTRL	65	P2[3]	P3[3]
Logger_UART_RX_in	66	P2[4]	P3[4]
Atmel_MOSI	68	P2[5]	P3[5]
BLE_BNO_RST	1	P2[6]	P3[6]
BLE_P10_5	2	P2[7]	P3[7]
SYS_SCL	38	P12[0]	P15[0]
SYS_SDA	39	P12[1]	P15[1]
KP_SWDIO	46	P12[2]	P15[2]
KP_SWDCLK	47	P12[3]	P15[3]
KP_SWDRST	3	P12[4]	P15[4]
Atmel_MISO	4	P12[5]	P15[5]
BLE_scb5_RX_in	20	P12[6]	P15[6]
BLE_scb5_TX_out	21	P12[7]	P15[7]
IND	6		
PSoC5_SWDIO	11		
PSoC5_SWDCLK	12		
EC_INT	13		
F3_BOOT0	14		
KP_P1_4	15		
L/R_SELECT	16		
KP_P1_6	18		
KP_P1_7	19		
F4_UART1_TX	29		
F4_BOOT0	30		
F3_nRST	31		
F3_TX1_out	32		
F4_UART1_RX	33		
F4_UART3_TX	34		
F4_UART3_RX	36		
F4_UART4_TX	37		
F4_UART4_RX	27		
F4_UART6_TX	28		
F4_UART6_RX	40		
BLE_TX_out	41		
BLE_RX_in	60		
BLE_P10_6	61		
PSoC_DP_P_P	22		
PSoC_DP_P_N	23		

3p3V — PU for KitProg R16 24K — Logger_UART_TX_out
R47 24K — Atmel_SCK
KP_P0_2
KP_P0_3
KP_P0_4
F4_nRST
F3_RX1_in
F4_PC4
R59 24K — Atmel_RST
R60 100K — BNO_RST
USB_BRIDGE_nRST
SD_CARD_nPWR_CTRL
Logger_UART_RX_in
Atmel_MOSI
BLE_BNO_RST
BLE_P10_5
SYS_SCL
SYS_SDA
KP_SWDIO
KP_SWDCLK
KP_SWDRST
Atmel_MISO
BLE_scb5_RX_in
BLE_scb5_TX_out
DGND

PB1 SPST-NO — R42 1.5K — D7 Green — R5 330 — DGND
R9 24K — PU for KitProg — 3p3V
R61 100K
R6 1.5K — DGND
R62 100K — USB_DS_D_P
R31 22 — USB_DS_D_P
R32 22 — USB_DS_D_N — DGND

For KitProg Operation:
- P3[2] and P0[5] pulled low
- P3[3], P0[7], P0[6], and P0[4] should float

25 mA Max @3.3V

U6B CY8C5868LTI-LP039

VBOOST	7	VDDIO0	52
VBAT	8	VDDIO1	17
		VDDIO2	67
		VDDIO3	35
		EP	69
VCCA	42	VSSA	43
VDDA	44		
		VSSB	5
VCCD	26	VSSD	9
VCCD	57	VSSD	25
VDDD	24	VSSD	45
VDDD	59	VSSD	58

3p3V — L2 200mA Ferrite
C27 1µF, C28 1µF, C29 1µF, C30 0.1µF
C33 1µF, C34 0.1µF, C35 1µF, C36 0.1µF
C37 1µF, C38 1µF
C25 0.1µF, C26 0.1µF, C31 0.1µF, C32 0.1µF
DGND

U10 N25Q512A13GF840E

KP_P1_6	6	C	
KP_P1_7	5	DQ0	W#/VPP/DQ2
KP_P1_4	2	DQ1	HOLD#/DQ3
			S#
			EP
	8	VCC	VSS

S#	1
W#/VPP/DQ2	3
HOLD#/DQ3	7
EP	9
VSS	4

KP_P0_4 — R18 12K — R19 12K — 3p3V
3p3V — C5 1µF — DGND

Test Points

3p3V — J3 HEADER 1x6 50mil
KP_P0_3
KP_P1_7
KP_P1_6
KP_P0_2
D21 3.3V, D23 3.3V, D20 3.3V, D22 3.3V — DGND

SWD to Program PS0C5

PSoC5_SWDIO
PSoC5_SWDCLK
PSoC5_SWDRST
3p3V — J4 — DGND

SIGNAL COLOR LEGEND
BLE<-->F4 SIGNALS
BLE<-->PSoC5 SIGNALS
F4<-->PSoC5 SIGNALS

F4_nRST
F4_BOOT0
L/R_SELECT
F4_UART1_TX
F4_UART1_RX
F4_UART2_TX
F4_UART2_RX
F4_UART3_TX
F4_UART3_RX
F4_UART4_TX
F4_UART4_RX
F4_UART6_TX
F4_UART6_RX
F4_PB13
1-WIRE
F4_PC1
F4_PC2
F4_PC3
F4_PC4
F4_PC13
F4_BNO_nRST

F4_I/O

UART1: Sniffer
UART2: BNO serial option
UART3: BLE and Serial-Bootloading
UART4: Motor Controller COMMS
UART6: Console

The PSoC5 has access to, and can route, all of these UART interfaces from the F4, so different interfaces can be logged on the serial datalogger, as well as real-time viewing via USB-UART-Bridging The PSoC5 also controls the F4 nRST and BOOT0 pins to be able to bootload the F4 over UART.

The PSoC5 also drives the L/R Select line to the F4 to tell it which shoe it is, and also forwards the nRST signal to the BNO.

USB_DS_D_P
USB_DS_D_N

USB_DnStr

USB is used for KitProg-programming of BLE, and it is used for USB-UART-bridging to bootload the F4 and the F3.

F3_nRST
F3_BOOT0
F3_TX1_out
F3_RX1_in

F3_I/O

The PSoC5 forwards the UART traffic from either the F4 or the M4, to the F3. It also has access to the F3 nRST and BOOT0 pins, so that it can bootload the F3 via USB-UART-bridging.

SYS_SCL
SYS_SDA

SYS_I2C

The PSoC5 is connected to the System-Wide I2C bus to enable communications between the MCU-board, and the USBPD-board.

BLE_TX_out
BLE_RX_in
BLE_scb5_TX_out
BLE_scb5_RX_in
BLE_BNO_RST
BLE_P10_5
BLE_P10_6

BLE_I/O

The PSoC5 forwards UART traffic between F4 and BLE.

The 2nd BLE UART connection, and other GPIO are for expandability.

Revision Updates

Title: *mk8p0 MCU*		
Size: B	Number: 4	Revision: REV-2
Date: 6/17/2020	Time: 3:20:05 PM	Sheet 4 of 7
File: C:\Temp\Delete\Nimbus\mk8p0_MCU\PSoC5LP.SchDoc		

Nimbus Robotics
4620 Henry St
Pittsburgh, PA 15213

Nimbus Robotics

50 mA Max @3.3V

Revision Updates

KP_SWD
KP_SWDIO
KP_SWDCLK
KP_SWDRST
KP_SWDIO
KP_SWDCLK
KP_SWDRST

These are the KitProg signals from PSoC5 for reprogramming the BLE module in-system via USB.

BLE_I/O
BLE_TX_out
BLE_RX_in
BLE_scb5_TX_out
BLE_scb5_RX_in
BLE_TX_out
BLE_RX_in
BLE_scb5_TX_out
BLE_scb5_RX_in
BLE_BNO_RST
BLE_P10_5
BLE_P10_6
BLE_BNO_RST
BLE_P10_5
BLE_P10_6

F4_I/O
F4_nRST
F4_BOOT0
L/R_SELECT
F4_UART1_TX
F4_UART1_RX
F4_UART2_TX
F4_UART2_RX
F4_UART3_TX
F4_UART3_RX
F4_UART4_TX
F4_UART4_RX
F4_UART6_TX
F4_UART6_RX
F4_PB13
1-WIRE
F4_PC1
F4_PC2
F4_PC3
F4_PC4
F4_PC13
F4_nRST
F4_BOOT0
L/R_SELECT
F4_UART1_TX
F4_UART1_RX
F4_UART2_TX
F4_UART2_RX
F4_UART3_TX
F4_UART3_RX
F4_UART4_TX
F4_UART4_RX
F4_UART6_TX
F4_UART6_RX
F4_PB13
1-WIRE
F4_PC1
F4_PC2
F4_PC3
F4_PC13
UART3_CTS
TP1
BT_MOD_TP3
DBG_LED_4
F4_BNO_nRST
BT_MOD_TP4

BNO_INT
BNO_INT

BNO_I2C
BNO_SCL
BNO_SDA
BNO_SCL
BNO_SDA

SYS_I2C
SYS_SCL
SYS_SDA
SYS_SCL
SYS_SDA

X3
1
2
32.768kHz
C39
22pF
C40
22pF
DGND
DGND
3p3V
C42
2.2µF
C43
0.1µF
DGND
DGND
DGND
BLE_P0_5
M1_CTS

U7
CYBLE-416045-02

1	GND1	GND3	43
2	P0.5	XRES	42
3	VBACK_UP	P0.4	41
4	VDD	P5.1	40
5	P0.0	P5.0	39
6	P0.1	P5.2	38
7	P10.3	P5.3	37
8	P10.4	P5.5	36
9	P9.3	P12.7	35
10	P10.6	P12.6	34
11	P10.5	P10.2	33
12	P10.1	P5.6	32
13	P10.0	P7.7	31
14	P9.4	P6.3	30

F4_PB13
BLE_BNO_RST
1-WIRE
BLE_P10_6
BLE_P10_5
BLE_TX_out
BLE_RX_in
F4_PC1
KP_SWDRST
F4_UART2_TX
BLE_scb5_TX_out
BLE_scb5_RX_in
F4_UART2_RX
32kHz_CLK
BLE_P5_5
BLE_P12_7
25MHz_CLK
BLE_P10_2
L/R_SELECT
F4_PC2
BT_MOD_TP3
DGND
32kHz_CLK
32kHz_CLK
25MHz_CLK
25MHz_CLK
TX2

15	GND2	P6.5	29
16	VREF	P6.2	28
17	P9.0	P6.1	27
18	P9.1	P6.7	26
19	P9.5	P5.4	25
20	P9.6	P6.4	24
21	P9.2	P7.1	23
22	P7.2	P7.2	22

BNO_SCL
BNO_SDA
GRN_LED
BLU_LED
WHI_LED
F4_PC3
F4_PC13
SYS_SCL
BNO_INT
KP_SWDCLK
KP_SWDIO
RED_LED
SYS_SDA
BT_MOD_TP4
DGND

50 mA Max @3.3V

4/21, idea for adding 3rd UART
- In order to accommodate a 3rd UART on BLE module for Boris: 1 sniffer, 1 console (M4 or M0), 1 UART to talk to F3:
- remove SYS_I2C port and replace with a UART
- connect new UART to UART2 of F4 (already on this page, just swap pins)
- F4 can re-route signal to PSoC5 for external exposure
- probably want to also connect this new UART to TPs

SWD
to Program PS0C6
KP_SWDIO
KP_SWDCLK
KP_SWDRST
3p3V
J8
DGND

Test Points
BLE_P5_5 R25
BLE_P12_7 R26
BLE_P10_2 R28
BLE_P0_5 R34
1.5K
1.5K
1.5K
1.5K
D24
3.3V
D27
3.3V
D25
3.3V
D26
3.3V
J5
DGND

RED_LED D6 R36 330 Red
BLU_LED D8 R38 330 Blue
WHI_LED D11 R41 330 White
GRN_LED D13 R44 330 Green
DGND

SIGNAL COLOR LEGEND
BLE<-->F4 SIGNALS
BLE<-->PSoC5 SIGNALS
F4<-->PSoC5 SIGNALS

Title	mk8p0 MCU		
Size: B	Number:5	Revision:REV-2	
Date: 6/17/2020	Time: 3:20:06 PM	Sheet 5 of 7	
File: C:\Temp\Delete\Nimbus\mk8p0_MCU\BLE.SchDoc			

Nimbus Robotics
4620 Henry St
Pittsburgh, PA 15213

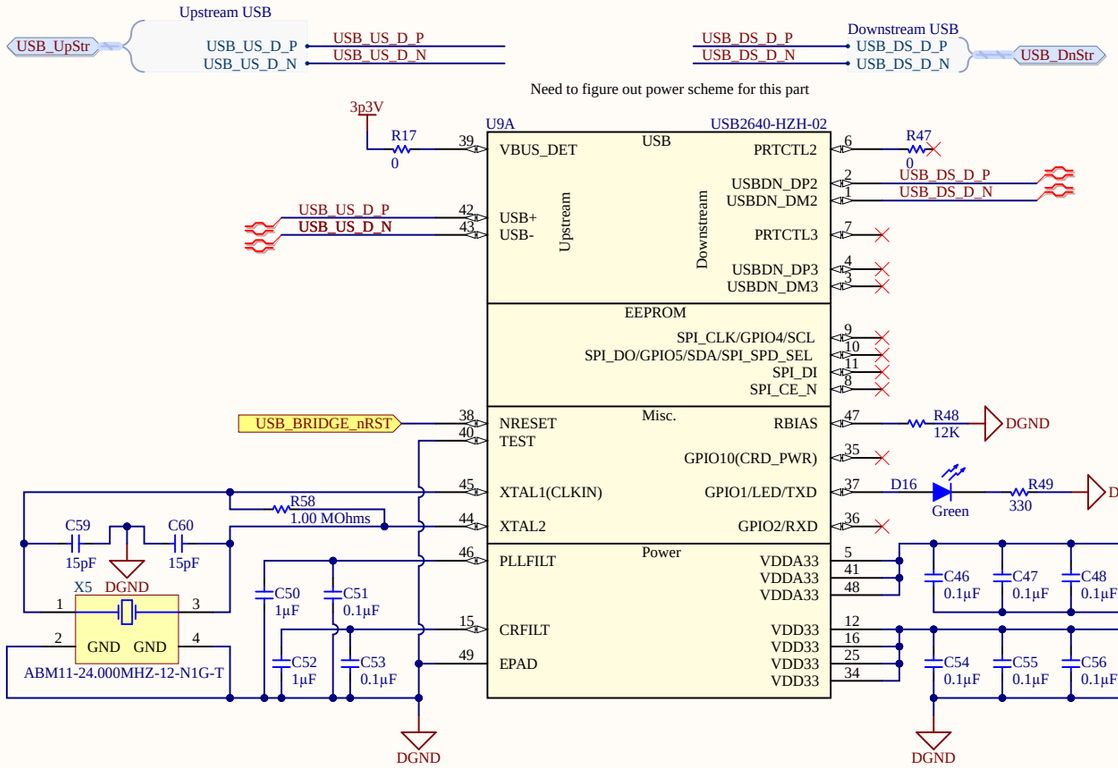
Nimbus Robotics

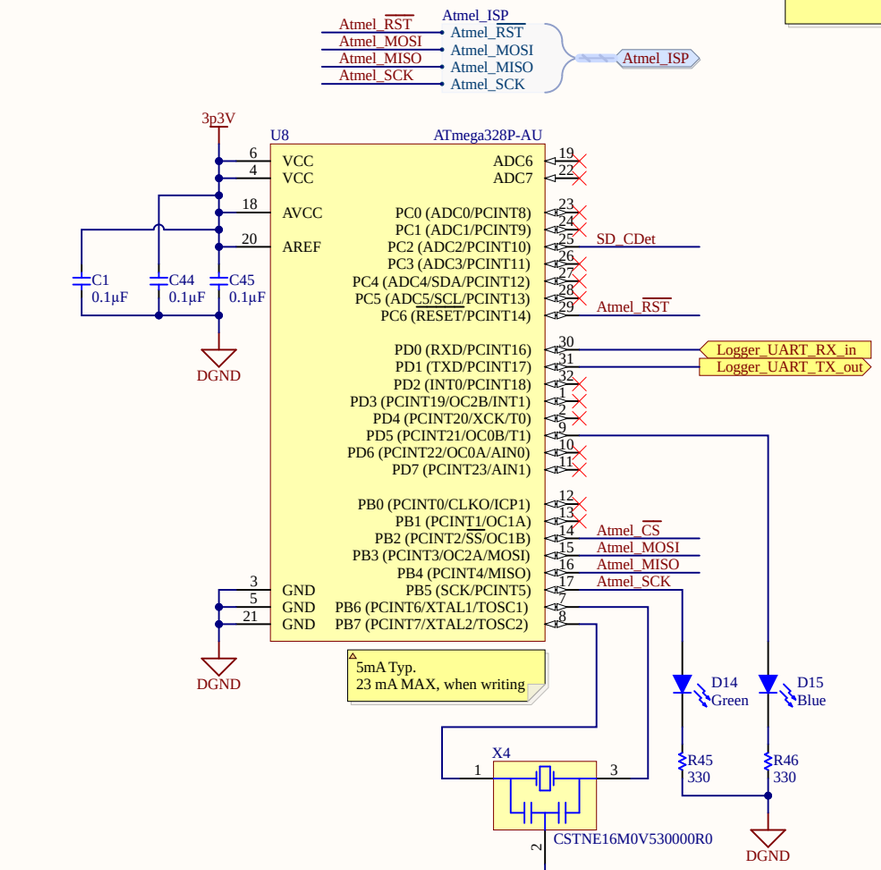
448 mA Max @3.3V

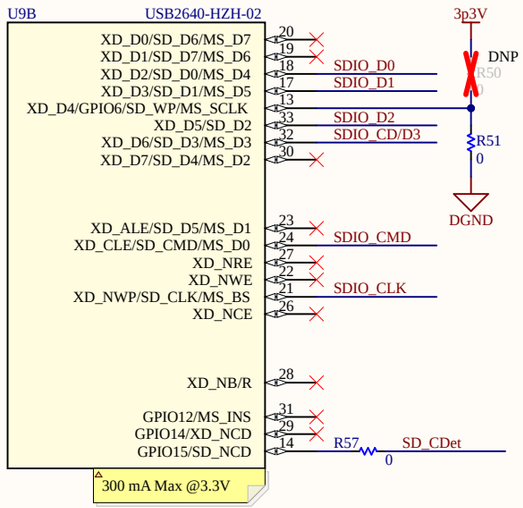
PSoC5 controls bridge-chip-reset, Atmel reset, and power to SD card. Using these signals it controls which chip can access the card, and which protocol is used (SDIO or SPI).

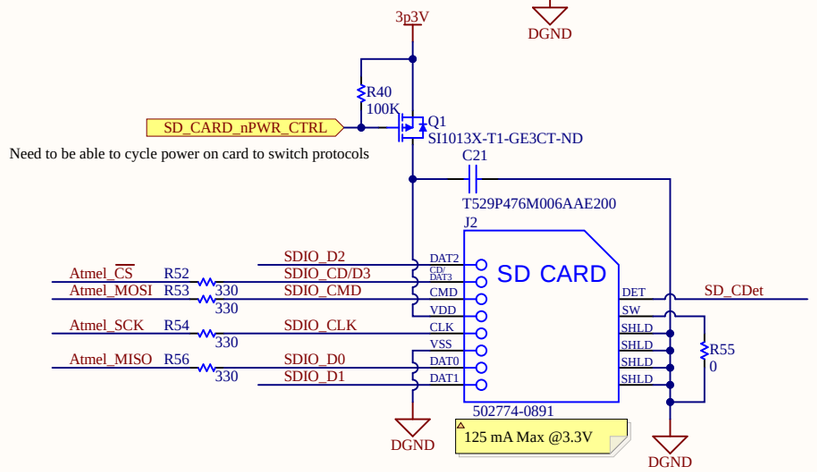
PSoC5 is notified by USB-PD controller when USB is connected, and then enables the bridge-chip, and disables Atmel.

Revision Updates





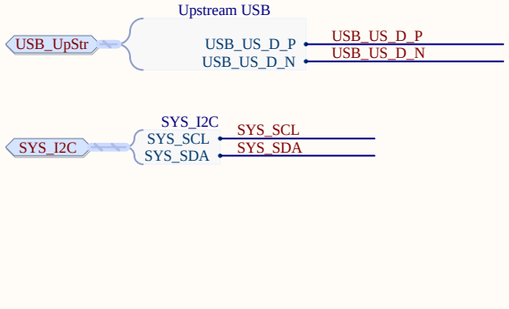


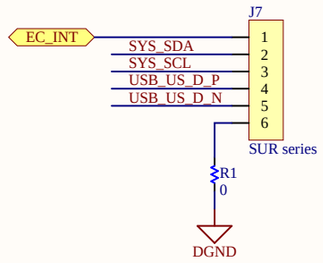


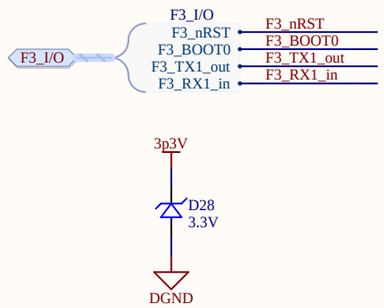
Title	***mk8p0 MCU***		*Nimbus Robotics* *4620 Henry St* *Pittsburgh, PA 15213*
Size: B	Number:6	Revision:REV-2	Nimbus Robotics
Date: 6/17/2020	Time: 3:20:06 PM	Sheet 6 of 7	
File: C:\Temp\Delete\Nimbus\mk8p0_MCU\DataLogger.SchDoc			

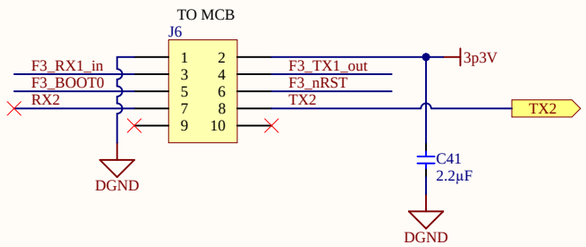
Revision Updates

PSB Comms









Title	***mk8p0 MCU***		*Nimbus Robotics* *4620 Henry St* *Pittsburgh, PA 15213*
Size: B	Number: 7	Revision: REV-2	Nimbus Robotics
Date: 6/17/2020	Time: 3:20:06 PM	Sheet 7 of 7	
File: C:\Temp\Delete\Nimbus\mk8p0_MCU\External_Conns.SchDoc			