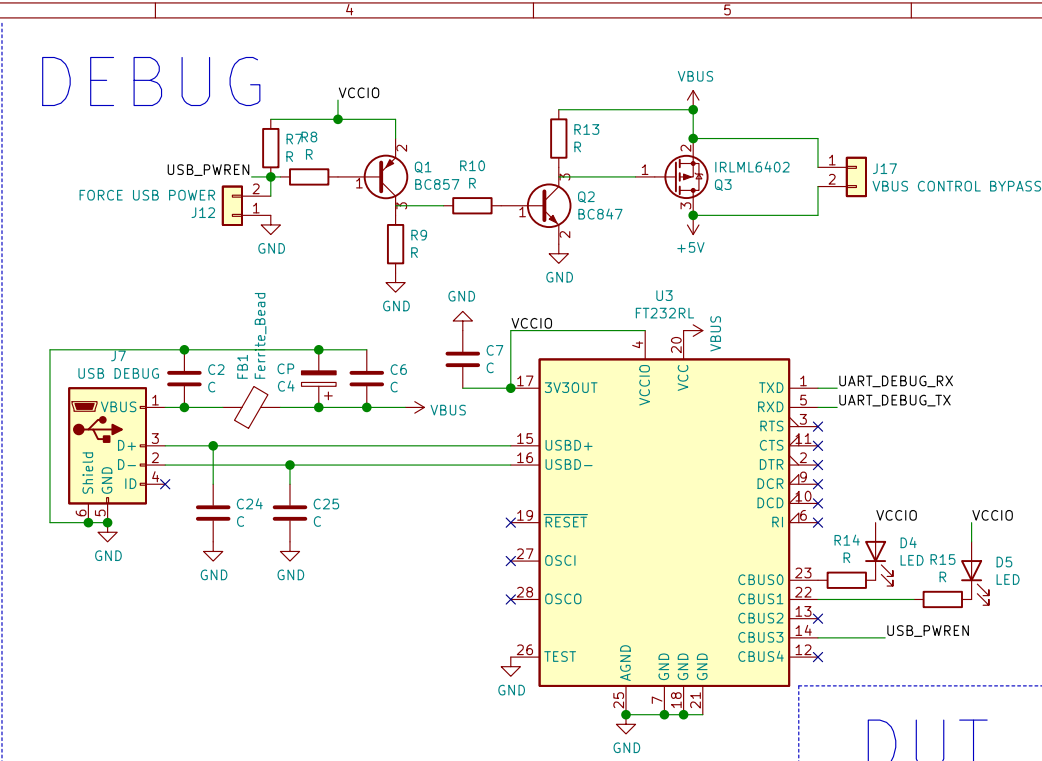
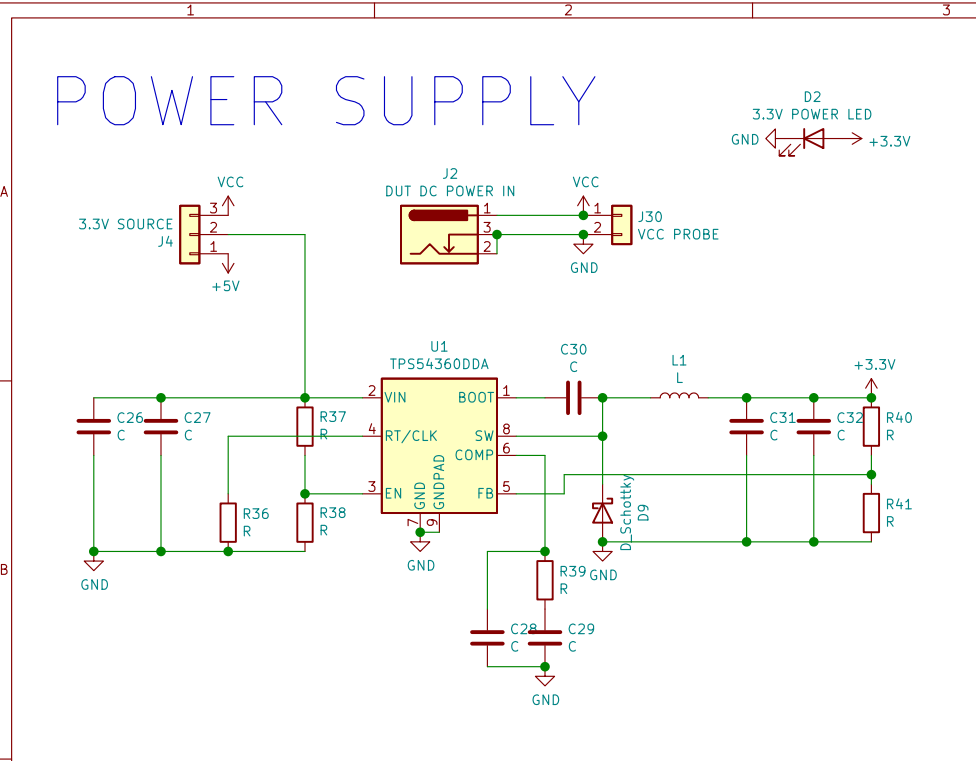
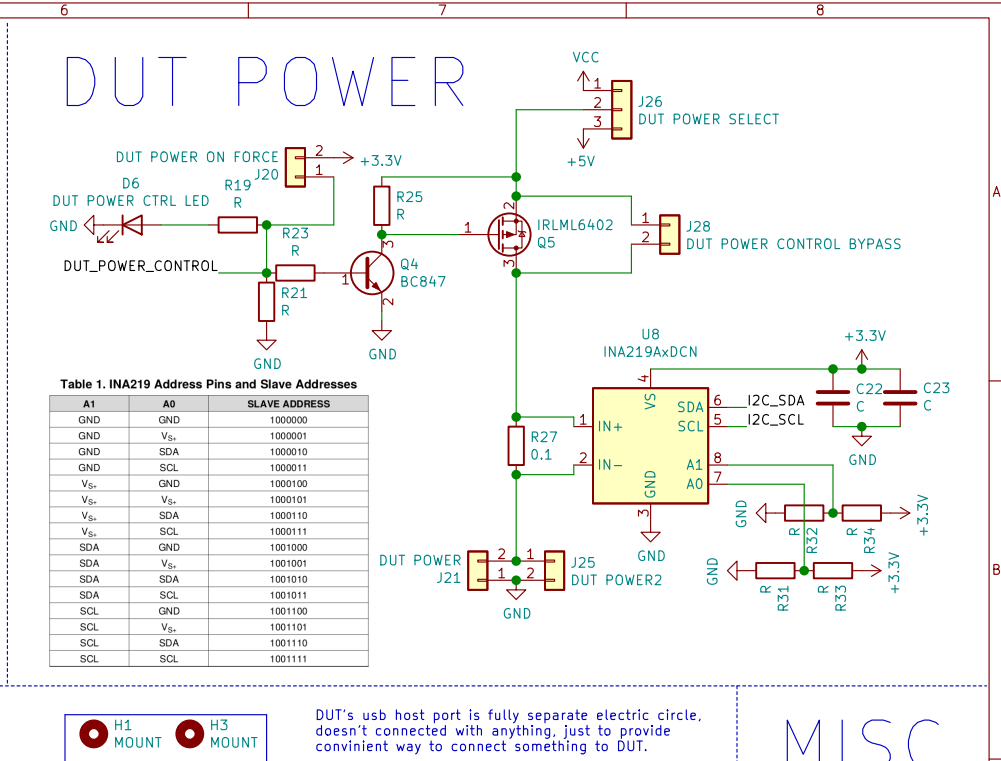


POWER SUPPLY

The diagram illustrates a 3.3V power supply circuit. It features a TPS54360DDA (U1) converter. The input is a 3.3V source (J4) connected to the VIN pin (pin 2). The output is a 3.3V power LED (D2) connected to the SW pin (pin 8) through an inductor (L1). The circuit includes various passive components: capacitors C26, C27, C30, C31, C32, C28, and C29; resistors R36, R37, R38, R39, R40, and R41; and a Schottky diode D9. The output voltage is labeled as +3.3V. A 3.3V power LED (D2) is shown at the top right, connected to GND and +3.3V. A DUT DC power input (J2) is also shown, connected to the VIN pin (pin 2) and GND.

[illegible]

The diagram illustrates the power management for the DUT (Device Under Test). It features a +5V input connected to a J26 connector (DUT POWER SELECT) and a +3.3V input connected to a J28 connector (DUT POWER CONTROL BYPASS). The +5V line passes through a resistor R25 and a diode Q5 (IRLML6402) to a J25 connector (DUT POWER2). The +3.3V line is connected to a J21 connector (DUT_POWER_CONTROL) and a J26 connector (DUT POWER ON FORCE). The J21 connector is connected to a diode D6 (DUT POWER CTRL LED) and a resistor R23. The J26 connector is connected to a resistor R21 and a diode Q4 (BC847). The output of the diode Q4 is connected to a resistor R25 and a diode Q5 (IRLML6402). The output of the diode Q5 is connected to a resistor R27 (0.1) and a diode Q5 (IRLML6402). The output of the diode Q5 is connected to a resistor R27 (0.1) and a diode Q5 (IRLML6402). The output of the diode Q5 is connected to a resistor R27 (0.1) and a diode Q5 (IRLML6402).

Table 1. INA219 Address Pins and Slave Addresses

| A1 | A0 | SLAVE ADDRESS |
|-----------------|-----------------|---------------|
| GND | GND | 1000000 |
| GND | V _{SS} | 1000001 |
| GND | SDA | 1000010 |
| GND | SCL | 1000011 |
| V _{SS} | GND | 1000100 |
| V _{SS} | V _{SS} | 1000101 |
| V _{SS} | SDA | 1000110 |
| V _{SS} | SCL | 1000111 |
| SDA | GND | 1001000 |
| SDA | V _{SS} | 1001001 |
| SDA | SDA | 1001010 |
| SDA | SCL | 1001011 |
| SCL | GND | 1001100 |
| SCL | V _{SS} | 1001101 |
| SCL | SDA | 1001110 |
| SCL | SCL | 1001111 |

Camera Debug Facility

Device represents hands free remote embedded development conception.

Typical DUT has debug IART port, spi nor (rarely nand) flash ROM and Fast Ethernet. Software stack is U-Boot + Linux based OS.

During debug developer usually many times completely reupload full ROM image, use UART as main control and network connection on last stages of software development.

Usually such process involves DUT on developer's table. This device eliminates this requirement. Developer and DUT can be placed anywhere in separate locations connected via Internet.

Implementation tuned to satisfy basic needs for
typical 38x38mm cctv ip camera module development/debug.

Camera Debug Facility

Device represents hands free remote embedded development conception.

Typical DUT has debug IART port, spi nor (rarely nand) flash ROM and Fast Ethernet. Software stack is U-Boot + Linux based OS.

During debug developer usually many times completely reupload full ROM image, use UART as main control and network connection on last stages of software development.

Usually such process involves DUT on developer's table. This device eliminates this requirement. Developer and DUT can be placed anywhere in separate locations connected via Internet.

Implementation tuned to satisfy basic needs for
typical 38x38mm cctv ip camera module development/debug.

Camera Debug Facility

Device represents hands free remote embedded development conception.

Typical DUT has debug IART port, spi nor (rarely nand) flash ROM and Fast Ethernet. Software stack is U-Boot + Linux based OS.

During debug developer usually many times completely reupload full ROM image, use UART as main control and network connection on last stages of software development.

Usually such process involves DUT on developer's table. This device eliminates this requirement. Developer and DUT can be placed anywhere in separate locations connected via Internet.

Implementation tuned to satisfy basic needs for
typical 38x38mm cctv ip camera module development/debug.

Camera Debug Facility

Device represents hands free remote embedded development conception.

Typical DUT has debug IART port, spi nor (rarely nand) flash ROM and Fast Ethernet. Software stack is U-Boot + Linux based OS.

During debug developer usually many times completely reupload full ROM image, use UART as main control and network connection on last stages of software development.

Usually such process involves DUT on developer's table. This device eliminates this requirement. Developer and DUT can be placed anywhere in separate locations connected via Internet.

Implementation tuned to satisfy basic needs for
typical 38x38mm cctv ip camera module development/debug.

Camera Debug Facility

Device represents hands free remote embedded development conception.

Typical DUT has debug IART port, spi nor (rarely nand) flash ROM and Fast Ethernet. Software stack is U-Boot + Linux based OS.

During debug developer usually many times completely reupload full ROM image, use UART as main control and network connection on last stages of software development.

Usually such process involves DUT on developer's table. This device eliminates this requirement. Developer and DUT can be placed anywhere in separate locations connected via Internet.

Implementation tuned to satisfy basic needs for
typical 38x38mm cctv ip camera module development/debug.

Camera Debug Facility

Device represents hands free remote embedded development conception.

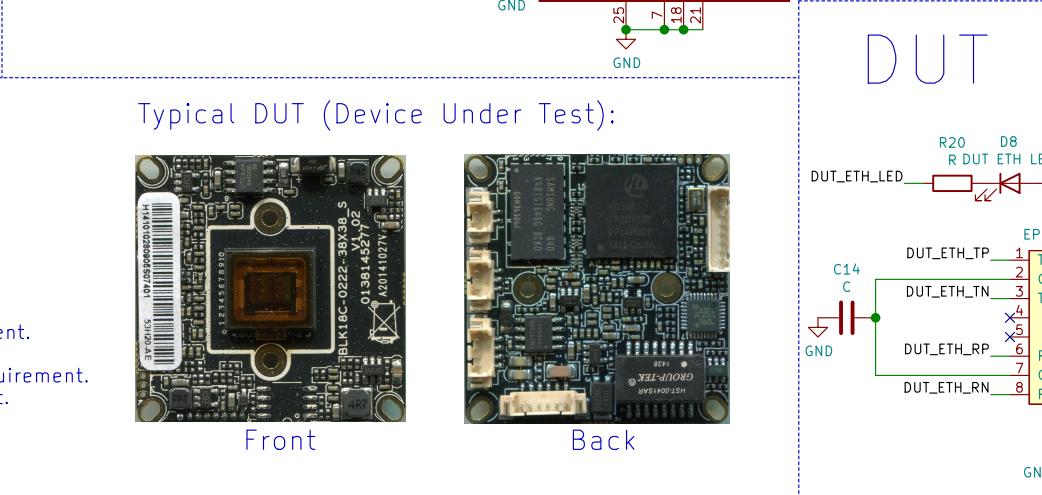
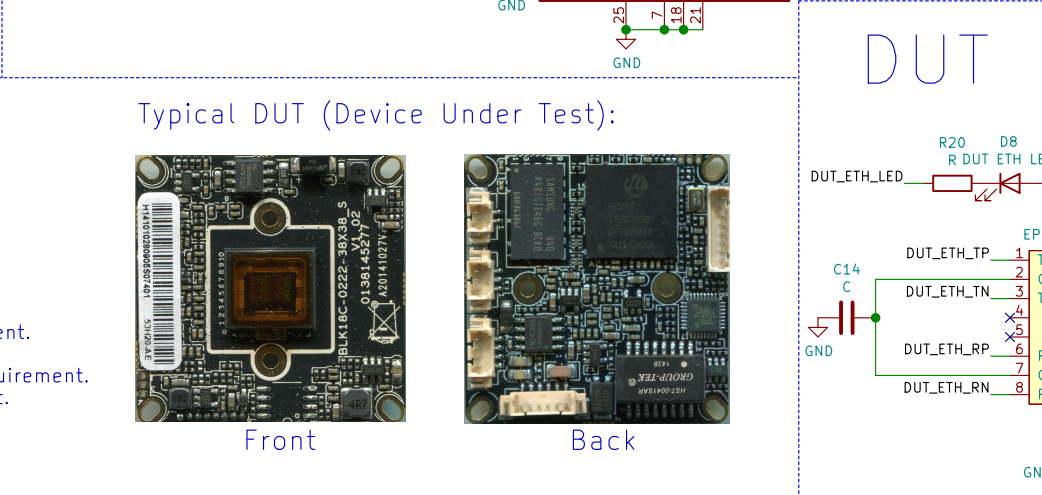
Typical DUT has debug IART port, spi nor (rarely nand) flash ROM and Fast Ethernet. Software stack is U-Boot + Linux based OS.

During debug developer usually many times completely reupload full ROM image, use UART as main control and network connection on last stages of software development.

Usually such process involves DUT on developer's table. This device eliminates this requirement. Developer and DUT can be placed anywhere in separate locations connected via Internet.

Implementation tuned to satisfy basic needs for
typical 38x38mm cctv ip camera module development/debug.

The figure illustrates a typical Device Under Test (DUT) setup. It includes two photographs of the device's front and back views, showing various components like chips and connectors. Accompanying schematics show how the device is interfaced with external systems. The top schematic shows a power supply configuration where pins 25, 7, 18, and 21 are grounded. The bottom schematic details signal connections: an LED (R20) for Ethernet status, a capacitor (C14), and four signal lines (TP, TN, RP, RN) for Ethernet communication.

[illegible]

The figure illustrates a typical Device Under Test (DUT) setup. It includes two photographs of the device's front and back views, showing various components like chips and connectors. Accompanying schematics show how the device is interfaced with external systems. The top schematic shows a power supply configuration where pins 25, 7, 18, and 21 are grounded. The bottom schematic details signal connections: an LED (R20) for Ethernet status, a capacitor (C14), and four signal lines (TP, TN, RP, RN) for Ethernet communication.

DUT

38*38mm mount holes

H1 MOUNT H3 MOUNT
H2 MOUNT H4 MOUNT

DUT_ETH_LED → R20 R DUT ETH LED → +3.3V

U6 EPC05-A8065

DUT_ETH_TP 1 TD+ TX+ 16
DUT_ETH_TN 2 CT CT 15
DUT_ETH_RP 4 RD+ RX+ 13
DUT_ETH_RN 8 RD- RX- 9

J19 DUT FETH

J23 DUT UART PROBE

J24 DUT UART

J27 DUT USB

J22 DUT USB PORT

VBUS 1
D+ 3
D- 2
GND 4

UART_DUT_RX
UART_DUT_TX

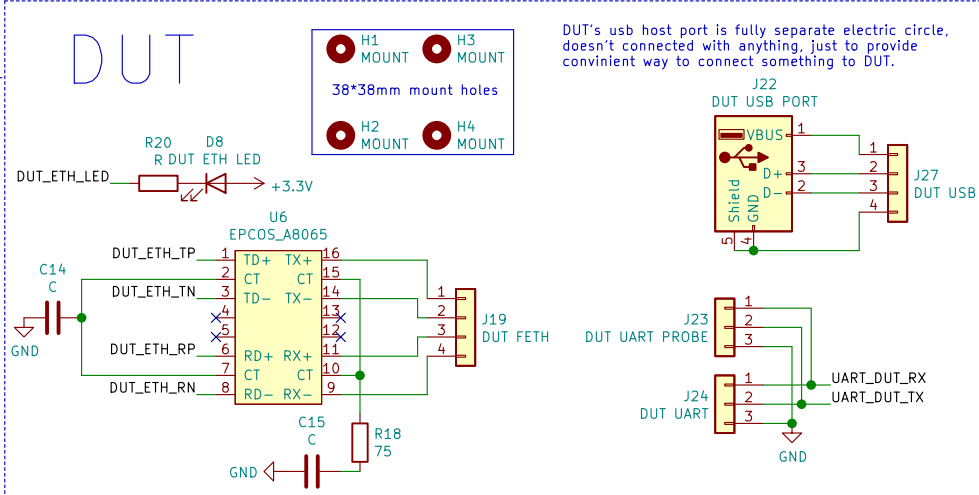
GND

C14 C

C15 C

R18 75

DUT's usb host port is fully separate electric circle, doesn't connected with anything, just to provide convenient way to connect something to DUT.



MISC

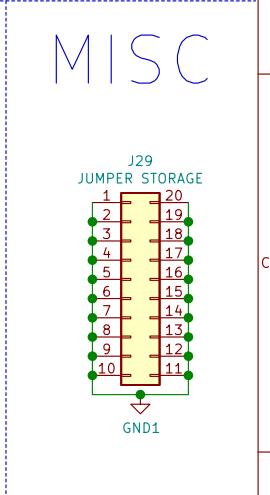
J29

JUMPER STORAGE

The diagram shows a 20-pin connector labeled J29. The pins are arranged in two columns of 10. The left column is numbered 1 to 10 from top to bottom. The right column is numbered 20 to 11 from top to bottom. Each pin is represented by a green circle. A red line connects the bottom of pin 10 to the bottom of pin 11, indicating a jumper. Below the connector, a red arrow points to a label 'GND1'.

| Pin Number | Pin Number |
|------------|------------|
| 1 | 20 |
| 2 | 19 |
| 3 | 18 |
| 4 | 17 |
| 5 | 16 |
| 6 | 15 |
| 7 | 14 |
| 8 | 13 |
| 9 | 12 |
| 10 | 11 |

GND1



MISC

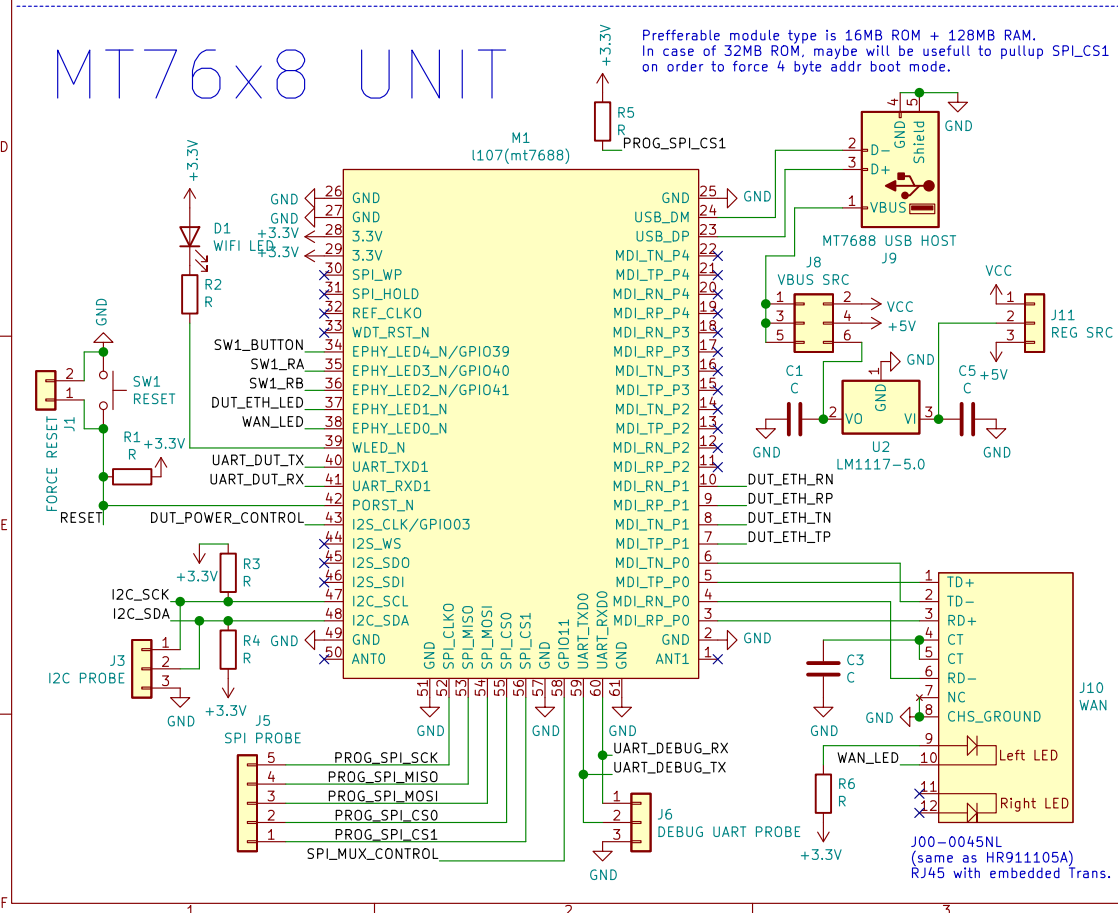
J29

JUMPER STORAGE

The diagram shows a 20-pin connector labeled J29. The pins are arranged in two columns of 10. The left column is numbered 1 to 10 from top to bottom. The right column is numbered 20 to 11 from top to bottom. Each pin is represented by a green circle. A red line connects the bottom of pin 10 to the bottom of pin 11, indicating a jumper. Below the connector, a red arrow points to a label 'GND1'.

| Pin Number | Pin Number |
|------------|------------|
| 1 | 20 |
| 2 | 19 |
| 3 | 18 |
| 4 | 17 |
| 5 | 16 |
| 6 | 15 |
| 7 | 14 |
| 8 | 13 |
| 9 | 12 |
| 10 | 11 |

GND1

[illegible][illegible]

SPI MUXER

Muxer by default connect SPI_FLASH to DUT_SPI_FLASH.
SPI_MUX_CONTROL High switches SPI_FLASH to PROG_SPI.
SPI_MUX_CONTROL is Low by default.

SN74CB3Q3257 functional modes:

| INPUTS | | INPUT/OUTPUT A | FUNCTION |
|--------|---|----------------|------------------|
| OE | S | A | |
| L | L | B1 | A port = B1 port |
| L | H | B2 | A port = B2 port |
| H | X | Z | Disconnect |

Muxer can be powered from DUT.
In this case, other components can be powered off.

DUT_SPI_VCC 1
SPLVCC 2 J13 DUT SPI FLASH PW SELECTOR
+3.3V ← 3

J14 DUT SPI FLASH
DUT_SPI_CS 1 8 DUT_SPI_VCC
DUT_SPI_MISO 2 7 DUT_SPI_HOLD
DUT_SPI_WP 3 6 DUT_SPI_SCK
4 5 DUT_SPI_MOSI
↓ GND

J15 SPI FLASH
SPLCS 1 8
SPLMISO 2 7
SPLWP 3 6
4 5 SPLMOSI
↓ GND

C8 C9
C C
GND SPLVCC

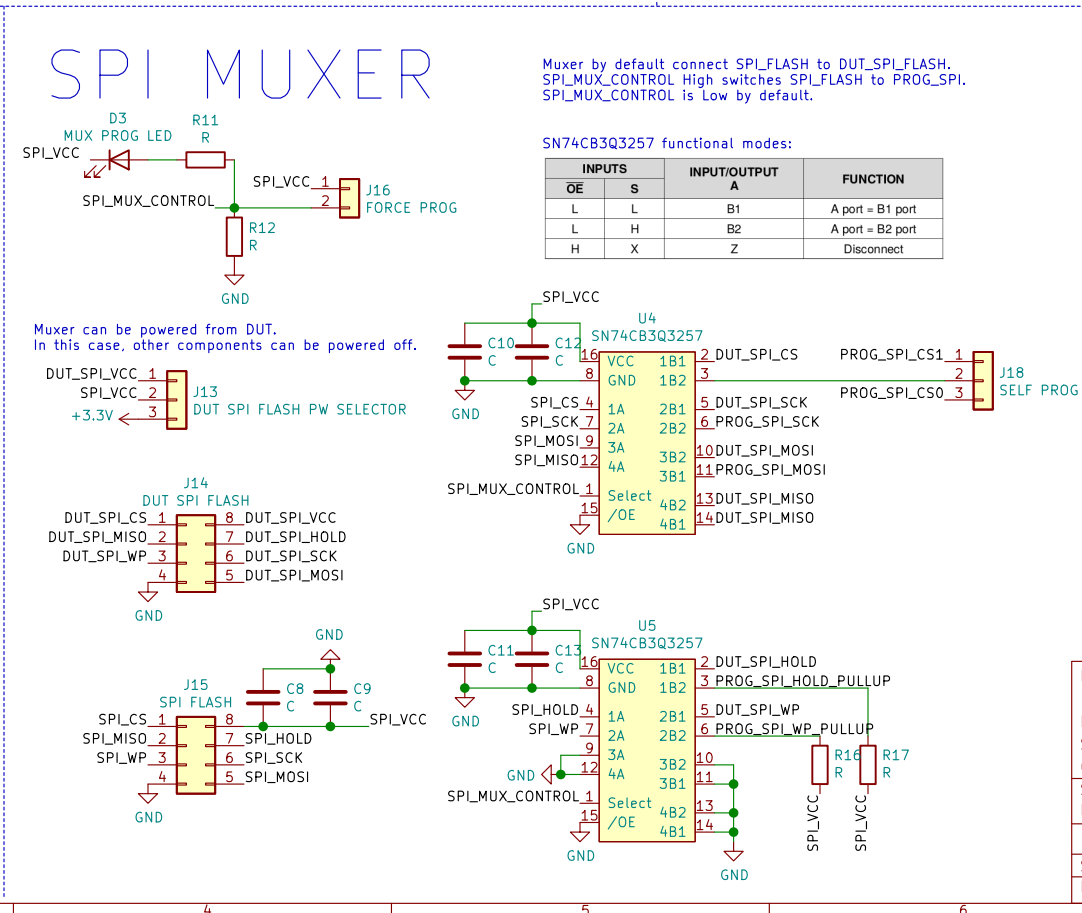
GND

SPLVCC
U4 SN74CB3Q3257
C10 C12
C C
GND
SPLCS 4 1A 2B1 5 DUT_SPI_CS
SPLSCK 7 2A 2B2 6 PROG_SPI_SCK
SPLMOSI 9 3A 3B2 10 DUT_SPI_MOSI
SPLMISO 12 4A 3B1 11 PROG_SPI_MOSI
SPL_MUX_CONTROL 1 Select 4B2 13 DUT_SPI_MISO
15 /OE 4B1 14 DUT_SPI_MISO
↓ GND

SPLVCC
U5 SN74CB3Q3257
C11 C13
C C
GND
SPL_HOLD 4 1A 2B1 5 DUT_SPI_WP
SPL_WP 7 2A 2B2 6 PROG_SPI_WP_PULLUP
12 3A 3B2 10
4A 3B1 11
Select 4B2 13
/OE 4B1 14
↓ GND

SPLVCC
R16 R
SPLVCC
R17 R

J18 SELF PROG
PROG_SPI_CS1 1 2
PROG_SPI_CS0 3



SPI MUXER

Muxer by default connect SPI_FLASH to DUT_SPI_FLASH.
SPI_MUX_CONTROL High switches SPI_FLASH to PROG_SPI.
SPI_MUX_CONTROL is Low by default.

SN74CB3Q3257 functional modes:

| INPUTS | | INPUT/OUTPUT A | FUNCTION |
|--------|---|----------------|------------------|
| OE | S | A | |
| L | L | B1 | A port = B1 port |
| L | H | B2 | A port = B2 port |
| H | X | Z | Disconnect |

Muxer can be powered from DUT.
In this case, other components can be powered off.

DUT_SPI_VCC 1
SPLVCC 2 J13 DUT SPI FLASH PW SELECTOR
+3.3V ← 3

J14 DUT SPI FLASH
DUT_SPI_CS 1 8 DUT_SPI_VCC
DUT_SPI_MISO 2 7 DUT_SPI_HOLD
DUT_SPI_WP 3 6 DUT_SPI_SCK
4 5 DUT_SPI_MOSI
↓ GND

J15 SPI FLASH
SPLCS 1 8
SPLMISO 2 7
SPLWP 3 6
4 5 SPLMOSI
↓ GND

C8 C9
C C
GND SPLVCC

GND

SPLVCC
R11 R
MUX PROG LED
SPLVCC
SPL_MUX_CONTROL
R12 R
↓ GND

SPL_VCC 1
FORCE PROG 2 J16

U4 SN74CB3Q3257
C10 C12
C C
GND
SPL_VCC 16
VCC 18B1 2 DUT_SPI_CS
GND 18B2 3
SPLCS 4 1A 28B1 5 DUT_SPI_SCK
SPLSCK 7 2A 28B2 6 PROG_SPI_SCK
SPLMOSI 9 3A 38B2 10 DUT_SPI_MOSI
SPLMISO 12 4A 38B1 11 PROG_SPI_MOSI
SPL_MUX_CONTROL 1 Select 48B2 13 DUT_SPI_MISO
15 /OE 48B1 14 DUT_SPI_MISO
↓ GND

U5 SN74CB3Q3257
C11 C13
C C
GND
SPL_VCC 16
VCC 18B1 2 DUT_SPI_HOLD
GND 18B2 3 PROG_SPI_HOLD_PULLUP
SPL_HOLD 4 1A 28B1 5 DUT_SPI_WP
SPL_WP 7 2A 28B2 6 PROG_SPI_WP_PULLUP
GND 9 3A 38B2 10
12 4A 38B1 11
SPL_MUX_CONTROL 1 Select 48B2 13
15 /OE 48B1 14
↓ GND

R16 R R17 R
SPLVCC SPLVCC

SPI MUXER

Muxer by default connect SPI_FLASH to DUT_SPI_FLASH.
SPI_MUX_CONTROL High switches SPI_FLASH to PROG_SPI.
SPI_MUX_CONTROL is Low by default.

SN74CB3Q3257 functional modes:

| INPUTS | | INPUT/OUTPUT A | FUNCTION |
|--------|---|----------------|------------------|
| OE | S | A | |
| L | L | B1 | A port = B1 port |
| L | H | B2 | A port = B2 port |
| H | X | Z | Disconnect |

Muxer can be powered from DUT.
In this case, other components can be powered off.

DUT_SPI_VCC 1
SPLVCC 2 J13 DUT SPI FLASH PW SELECTOR
+3.3V ← 3

J14 DUT SPI FLASH
DUT_SPI_CS 1 8 DUT_SPI_VCC
DUT_SPI_MISO 2 7 DUT_SPI_HOLD
DUT_SPI_WP 3 6 DUT_SPI_SCK
4 5 DUT_SPI_MOSI
↓ GND

J15 SPI FLASH
SPLCS 1 8
SPLMISO 2 7
SPLWP 3 6
4 5 SPLMOSI
↓ GND

C8 C9
C C
GND SPLVCC

GND

SPLVCC
U4 SN74CB3Q3257
C10 C12
C C
GND
SPLCS 4 1A 2B1 5 DUT_SPI_CS
SPLSCK 7 2A 2B2 6 PROG_SPI_SCK
SPLMOSI 9 3A 3B2 10 DUT_SPI_MOSI
SPLMISO 12 4A 3B1 11 PROG_SPI_MOSI
SPL_MUX_CONTROL 1 Select 4B2 13 DUT_SPI_MISO
15 /OE 4B1 14 DUT_SPI_MISO
↓ GND

SPLVCC
U5 SN74CB3Q3257
C11 C13
C C
GND
SPL_HOLD 4 1A 2B1 5 DUT_SPI_WP
SPL_WP 7 2A 2B2 6 PROG_SPI_WP_PULLUP
12 3A 3B2 10
4A 3B1 11
Select 4B2 13
/OE 4B1 14
↓ GND

SPLVCC
R16 R
SPLVCC
R17 R

J18 SELF PROG
PROG_SPI_CS1 1 2
PROG_SPI_CS0 3

USER INTERFACE

oled display(ssd1306)

U7

NC 30

C2P 29

C2N 28

C1P 27

C2N 26

VBAT 25

NC 24

VSS 23

VDD 22

BS0 21

BS1 20

BS2 19

CS# 18

RES# 17

E/RS# 16

D/C# 15

R/W# 14

NC 13

C2P 12

C2N 11

C1P 10

C2N 9

VBAT 8

NC 7

VSS 6

VDD 5

BS0 4

BS1 3

BS2 2

CS# 1

RES# 0

E/RS# 0

D/C# 0

R/W# 0

RESET

D7

R2

C16

C

GND

+3.3V

GND

R24

R

GND

+3.3V

R26

R

GND

+3.3V

GND

+3.3V

GND

C21

C

GND

R28

R

GND

+3.3V

GND

SW1_RA

R29

R

GND

+3.3V

SW1_RB

R30

R

GND

+3.3V

SW2

Encoder w BTN

A

C

B

S1

S2

GND

+3.3V

R35

R

GND

SW1_BUTTON

GND

+3.3V

For the SSD1306, the slave address is either "b0111100" or "b011101" by changing the SA0 to LOW or HIGH (D/C pin acts as SA0).

Hands free remote embedded development device.

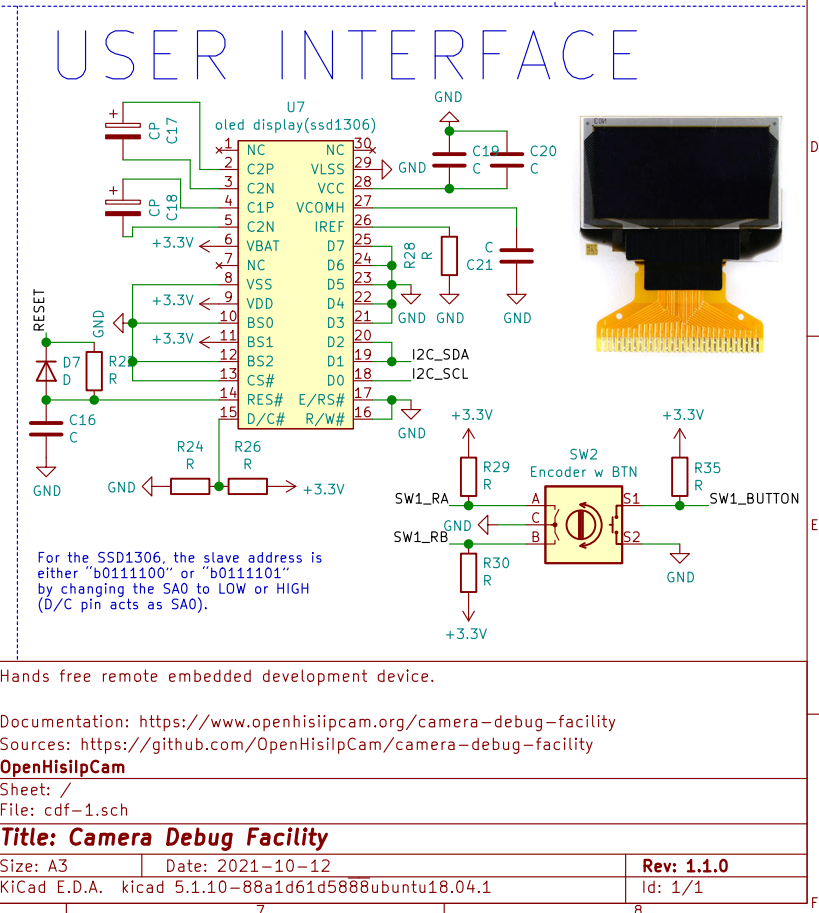
Documentation: <https://www.openhisiipcam.org/camera-debug-facility>
Sources: <https://github.com/OpenHisiIPCam/camera-debug-facility>

OpenHisiIPCam

Sheet: /
File: cdf-1.sch

Title: Camera Debug Facility

| | | |
|--|------------------|------------|
| Size: A3 | Date: 2021-10-12 | Rev: 1.1.0 |
| KiCad E.D.A. kicad 5.1.10-88a1d61d1804.1 | | Id: 1/1 |



USER INTERFACE

oled display(ssd1306)

For the SSD1306, the slave address is either "b0111100" or "b0111101" by changing the SA0 to LOW or HIGH (D/C pin acts as SA0).

Hands free remote embedded development device.

Documentation: <https://www.openhisiipcam.org/camera-debug-facility>
Sources: <https://github.com/OpenHisiIPCam/camera-debug-facility>

OpenHisiIPCam

Sheet: /
File: cdf-1.sch

Title: Camera Debug Facility

| | | |
|--|------------------|------------|
| Size: A3 | Date: 2021-10-12 | Rev: 1.1.0 |
| KiCad E.D.A. kicad 5.1.10-88a1d61d1804.1 | | Id: 1/1 |

Hands free remote embedded development device.

Documentation: <https://www.openhisiipcam.org/camera-debug-facility>
Sources: <https://github.com/OpenHisilpCam/camera-debug-facility>
OpenHisilpCam
Sheet: /
File: cdf-1.sch

| Title: Camera Debug Facility | | |
|---|------------------|-------------------|
| Size: A3 | Date: 2021-10-12 | Rev: 1.1.0 |
| KiCad E.D.A. kicad 5.1.10-88a1d61d5888ubuntu18.04.1 | | Id: 1/1 |

Hands free remote embedded development device.

Documentation: <https://www.openhisiipcam.org/camera-debug-facility>
Sources: <https://github.com/OpenHisilpCam/camera-debug-facility>
OpenHisilpCam
Sheet: /
File: cdf-1.sch

| Title: Camera Debug Facility | | |
|---|------------------|-------------------|
| Size: A3 | Date: 2021-10-12 | Rev: 1.1.0 |
| KiCad E.D.A. kicad 5.1.10-88a1d61d5888ubuntu18.04.1 | | Id: 1/1 |

Hands free remote embedded development device.

Documentation: <https://www.openhisiipcam.org/camera-debug-facility>
Sources: <https://github.com/OpenHisilpCam/camera-debug-facility>
OpenHisilpCam
Sheet: /
File: cdf-1.sch

| Title: Camera Debug Facility | | |
|---|------------------|-------------------|
| Size: A3 | Date: 2021-10-12 | Rev: 1.1.0 |
| KiCad E.D.A. kicad 5.1.10-88a1d61d5888ubuntu18.04.1 | | Id: 1/1 |

Hands free remote embedded development device.

Documentation: <https://www.openhisiipcam.org/camera-debug-facility>
Sources: <https://github.com/OpenHisilpCam/camera-debug-facility>
OpenHisilpCam
Sheet: /
File: cdf-1.sch

| Title: Camera Debug Facility | | |
|---|------------------|-------------------|
| Size: A3 | Date: 2021-10-12 | Rev: 1.1.0 |
| KiCad E.D.A. kicad 5.1.10-88a1d61d5888ubuntu18.04.1 | | Id: 1/1 |

Hands free remote embedded development device.

Documentation: <https://www.openhisiipcam.org/camera-debug-facility>
Sources: <https://github.com/OpenHisilpCam/camera-debug-facility>
OpenHisilpCam
Sheet: /
File: cdf-1.sch

| Title: Camera Debug Facility | | |
|---|------------------|-------------------|
| Size: A3 | Date: 2021-10-12 | Rev: 1.1.0 |
| KiCad E.D.A. kicad 5.1.10-88a1d61d5888ubuntu18.04.1 | | Id: 1/1 |

Hands free remote embedded development device.

Documentation: <https://www.openhisiipcam.org/camera-debug-facility>
Sources: <https://github.com/OpenHisilpCam/camera-debug-facility>
OpenHisilpCam
Sheet: /
File: cdf-1.sch

| Title: Camera Debug Facility | | |
|---|------------------|-------------------|
| Size: A3 | Date: 2021-10-12 | Rev: 1.1.0 |
| KiCad E.D.A. kicad 5.1.10-88a1d61d5888ubuntu18.04.1 | | Id: 1/1 |