

.....

Copyright 2018 Texas Instruments Incorporated. All rights reserved.

IMPORTANT: Your use of this Software is limited to those specific rights granted under the terms of a software license agreement between the user who downloaded the software, his/her employer (which must be your employer) and Texas Instruments Incorporated (the "License"). You may not use this Software unless you agree to abide by the terms of the License. The License limits your use, and you acknowledge, that the Software may not be modified, copied or distributed unless embedded on a Texas Instruments microcontroller which is integrated into your product. Other than for the foregoing purpose, you may not use, reproduce, copy, prepare derivative works of, modify, distribute, perform, display or sell this Software and/or its documentation for any purpose.

YOU FURTHER ACKNOWLEDGE AND AGREE THAT THE SOFTWARE AND DOCUMENTATION ARE PROVIDED AS IS WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY, TITLE, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL TEXAS INSTRUMENTS OR ITS LICENSORS BE LIABLE OR OBLIGATED UNDER CONTRACT, NEGLIGENCE, STRICT LIABILITY, CONTRIBUTION, BREACH OF WARRANTY, OR OTHER LEGAL EQUITABLE THEORY ANY DIRECT OR INDIRECT DAMAGES OR EXPENSES INCLUDING BUT NOT LIMITED TO ANY INCIDENTAL, SPECIAL, INDIRECT, PUNITIVE OR CONSEQUENTIAL DAMAGES, LOST PROFITS OR LOST DATA, COST OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY, SERVICES, OR ANY CLAIMS BY THIRD PARTIES (INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF), OR OTHER SIMILAR COSTS.

Should you have any questions regarding your right to use this Software, contact Texas Instruments Incorporated at [www.ti.com](http://www.ti.com).

.....

```
# OVT10640 reset and initialization
# Run from UB954
# Version 0.91
```

```
import time
import numpy as np
#reset and initialize OVT10640
```

```
# Set up IDs
UB954 = 0x60
UB953ID = 0x30
UB953 = 0x18
OVTID = 0x6C
OVT = 0x4C

# /* Reset Register, Digital Reset 0 */
board.WriteI2C(UB954,0x01,0x03)
time.sleep(1.0)

# /* Enable port 0 */
board.WriteI2C(UB954,0x4C,0x01)
time.sleep(1.0)

# /* Enable IIC */
board.WriteI2C(UB954,0x58,0x5E)

# /* Set 953/sensor address */
board.WriteI2C(UB954,0x5B,UB953ID)
time.sleep(1.0)
board.WriteI2C(UB954,0x5C,UB953)
time.sleep(1.0)
board.WriteI2C(UB954,0x5D,OVTID)
time.sleep(1.0)
board.WriteI2C(UB954,0x65,OVT)
time.sleep(1.0)

# /* RX port config */
board.WriteI2C(UB954,0x6D,0x7C)
time.sleep(1.0)

board.WriteI2C(UB954,0x32,0x01)
time.sleep(1.0)

# /* CSI_CTL Register: 4 lanes, enable CSI output */
board.WriteI2C(UB954,0x33,0x03) #0x21
time.sleep(1.0)

# /* FWD_CTL2 Register: best effort enabled */
board.WriteI2C(UB954,0x21,0x01)
time.sleep(1.0)
```

```
# /* FWD_CTL1 Register: enable RX Port 0 */
board.WriteI2C(UB954,0x20,0x20)
time.sleep(1.0)

# /* CSI rate: 800 Mbps serial rate */
board.WriteI2C(UB954,0x1F,0x02)
time.sleep(1.0)

# /* RX port specific register: No CSI-2 channel virtual mapping */
board.WriteI2C(UB954,0x72,0x00)
time.sleep(1.0)

# /* SCL Time register, set I2C Master SCL time */
board.WriteI2C(UB954,0x0A,0x7C)
time.sleep(1.0)
board.WriteI2C(UB954,0x0B,0x7C)
time.sleep(1.0)

# /* Receiver port control register, Enable Port 0 Receiver */
board.WriteI2C(UB954,0x0C,0x81)
time.sleep(1.0)

# /* Reset Register, Digital Reset 1 */
board.WriteI2C(UB953,0x01,0x03)
time.sleep(1.0)

# /* General Configuration: I2C 1.8 voltage, 4-lane configuration, Transmitter CRC, CSI-2
Continuous Clock. */
board.WriteI2C(UB953,0x02,0x73)
time.sleep(1.0)

# /* SCL Time: set I2C Master SCL High Time. */
board.WriteI2C(UB953,0x0B,0x7C)
time.sleep(1.0)
board.WriteI2C(UB953,0x0C,0x7C)
time.sleep(1.0)

# /* Set output clock: 24MHZ. */
board.WriteI2C(UB953,0x06,0x41)
time.sleep(1.0)
board.WriteI2C(UB953,0x07,0x28)
time.sleep(1.0)
```

```
print "OV2740 register"
```

```
reg_list = np.array([  
    [0x01, 0x03, 0x01],  
    [0x03, 0x02, 0x18],  
    [0x03, 0x0d, 0x1e],  
    [0x03, 0x0e, 0x02],  
    [0x03, 0x12, 0x01],  
    [0x03, 0x12, 0x01],  
    [0x30, 0x00, 0x00],  
    [0x30, 0x18, 0x32],  
    [0x30, 0x31, 0x0a],  
    [0x30, 0x80, 0x08],  
    [0x30, 0x83, 0xB4],  
    [0x31, 0x03, 0x00],  
    [0x31, 0x04, 0x01],  
    [0x31, 0x06, 0x01],  
    [0x35, 0x00, 0x00],  
    [0x35, 0x01, 0x44],  
    [0x35, 0x02, 0x40],  
    [0x35, 0x03, 0x88],  
    [0x35, 0x07, 0x00],  
    [0x35, 0x08, 0x00],  
    [0x35, 0x09, 0x80],  
    [0x35, 0x0c, 0x00],  
    [0x35, 0x0d, 0x80],  
    [0x35, 0x10, 0x00],  
    [0x35, 0x11, 0x00],  
    [0x35, 0x12, 0x20],  
    [0x36, 0x32, 0x00],  
    [0x36, 0x33, 0x10],  
    [0x36, 0x34, 0x10],  
    [0x36, 0x35, 0x10],  
    [0x36, 0x45, 0x13],  
    [0x36, 0x46, 0x81],  
    [0x36, 0x36, 0x10],  
    [0x36, 0x51, 0x0a],  
    [0x36, 0x56, 0x02],  
    [0x36, 0x59, 0x04],  
    [0x36, 0x5a, 0xda],  
    [0x36, 0x5b, 0xa2],  
])
```

[0x36, 0x5c, 0x04],  
[0x36, 0x5d, 0x1d],  
[0x36, 0x5e, 0x1a],  
[0x36, 0x62, 0xd7],  
[0x36, 0x67, 0x78],  
[0x36, 0x69, 0x0a],  
[0x36, 0x6a, 0x92],  
[0x37, 0x00, 0x54],  
[0x37, 0x02, 0x10],  
[0x37, 0x06, 0x42],  
[0x37, 0x09, 0x30],  
[0x37, 0x0b, 0xc2],  
[0x37, 0x14, 0x63],  
[0x37, 0x15, 0x01],  
[0x37, 0x16, 0x00],  
[0x37, 0x1a, 0x3e],  
[0x37, 0x32, 0x0e],  
[0x37, 0x33, 0x10],  
[0x37, 0x5f, 0x0e],  
[0x37, 0x68, 0x30],  
[0x37, 0x69, 0x44],  
[0x37, 0x6a, 0x22],  
[0x37, 0x7b, 0x20],  
[0x37, 0x7c, 0x00],  
[0x37, 0x7d, 0x0c],  
[0x37, 0x98, 0x00],  
[0x37, 0xa1, 0x55],  
[0x37, 0xa8, 0x6d],  
[0x37, 0xc2, 0x04],  
[0x37, 0xc5, 0x00],  
[0x37, 0xc8, 0x00],  
[0x38, 0x00, 0x00],  
[0x38, 0x01, 0x00],  
[0x38, 0x02, 0x00],  
[0x38, 0x03, 0x00],  
[0x38, 0x04, 0x07],  
[0x38, 0x05, 0x8f],  
[0x38, 0x06, 0x04],  
[0x38, 0x07, 0x43],  
[0x38, 0x08, 0x07],  
[0x38, 0x09, 0x80],  
[0x38, 0x0a, 0x04],  
[0x38, 0x0b, 0x38],  
[0x38, 0x0c, 0x05],

[0x38, 0x0d, 0x00],  
[0x38, 0x0e, 0x07],  
[0x38, 0x0f, 0x53],  
[0x38, 0x10, 0x00],  
[0x38, 0x11, 0x08],  
[0x38, 0x12, 0x00],  
[0x38, 0x13, 0x04],  
[0x38, 0x14, 0x01],  
[0x38, 0x15, 0x01],  
[0x38, 0x20, 0x86],  
[0x38, 0x21, 0x40],  
[0x38, 0x22, 0x84],  
[0x38, 0x29, 0x00],  
[0x01, 0x00, 0x01],  
[0x01, 0x00, 0x01],  
[0x01, 0x00, 0x01],  
[0x01, 0x00, 0x01],  
[0x01, 0x00, 0x01],  
[0x38, 0x2a, 0x01],  
[0x38, 0x2b, 0x01],  
[0x38, 0x30, 0x04],  
[0x38, 0x36, 0x01],  
[0x38, 0x37, 0x08],  
[0x38, 0x39, 0x01],  
[0x38, 0x3a, 0x00],  
[0x38, 0x3b, 0x08],  
[0x38, 0x3c, 0x00],  
[0x3f, 0x0b, 0x00],  
[0x40, 0x01, 0x20],  
[0x40, 0x09, 0x07],  
[0x40, 0x03, 0x10],  
[0x40, 0x10, 0xe0],  
[0x40, 0x16, 0x00],  
[0x40, 0x17, 0x10],  
[0x40, 0x44, 0x02],  
[0x43, 0x04, 0x08],  
[0x43, 0x07, 0x30],  
[0x43, 0x20, 0x80],  
[0x43, 0x22, 0x00],  
[0x43, 0x23, 0x00],  
[0x43, 0x24, 0x00],  
[0x43, 0x25, 0x00],  
[0x43, 0x26, 0x00],  
[0x43, 0x27, 0x00],

```
[0x43, 0x28, 0x00],
[0x43, 0x29, 0x00],
[0x43, 0x2c, 0x03],
[0x43, 0x2d, 0x81],
[0x45, 0x01, 0x84],
[0x45, 0x02, 0x40],
[0x45, 0x03, 0x18],
[0x45, 0x04, 0x04],
[0x45, 0x08, 0x02],
[0x46, 0x01, 0x10],
[0x48, 0x00, 0x00],
[0x48, 0x16, 0x52],
[0x48, 0x37, 0x1b],
[0x50, 0x00, 0x7f],
[0x50, 0x01, 0x00],
[0x50, 0x05, 0x38],
[0x50, 0x1e, 0x0d],
[0x50, 0x40, 0x00],
[0x59, 0x01, 0x00],
[0x01, 0x00, 0x01],
[0x01, 0x00, 0x01],
[0x01, 0x00, 0x01],
[0x01, 0x00, 0x01],
[0x01, 0x00, 0x01],
[0x35, 0x00, 0x00],
[0x35, 0x01, 0x46],
[0x35, 0x02, 0x60],
[0x35, 0x08, 0x00],
[0x35, 0x09, 0x10]
])
```

```
#Initialize OV2740
```

```
for i in range(0, 154):
```

```
    reg_1 = int(reg_list[i][0])
```

```
    reg_2 = int(reg_list[i][1])
```

```
    val = int(reg_list[i][2])
```

```
    board.Writel2C(OVT, reg_1, [reg_2, val])
```

```
#    print "SEND value: ", i, hex(reg_list[i][0]), hex(reg_list[i][1]), hex(reg_list[i][2])
```

```
# Seeing if CSI data is transmitting
```

```
print "THE END"
```