

## VCNL4000-BB

Version 1.1



# Table of Contents

---

1	Introduction.....	2
2	Board Features.....	2
3	Specifications.....	2
3.1	PCB Details.....	2
4	Hardware Connections.....	2
5	Pseudo Code.....	3
5.1	Register Addresses and Bit Values for Flags.....	3
5.2	Reading VCNL4000 Register.....	4
5.3	Writing VCNL4000 Register.....	4
5.4	Reading Ambient Light Value.....	5
5.5	Reading Proximity Distance Value.....	5
6	Reference.....	6



# 1 Introduction

VCNL4000-BB is a compact IR proximity and ambient light sensor breakout board. It can detect any obstacle within a range of about 20 cm. Proximity data and ambient light level data are read over I2C interface.

## 2 Board Features

- On-board 3.3V regulator
- Built-in ambient light sensor
- Adjustable IR LED pulse current

## 3 Specifications

- Input voltage: 5V
- IR LED pulse current: 10 mA - 200 mA  
Obstacle detection range: 20 cm
- Ambient light resolution: 0.25 lx
- I2C slave address: 0x26

### 3.1 PCB Details

- PCB type: FR4
- Solder mask: Black
- Board thickness: 1.6mm
- Surface finish: Immersion gold

## 4 Hardware Connections

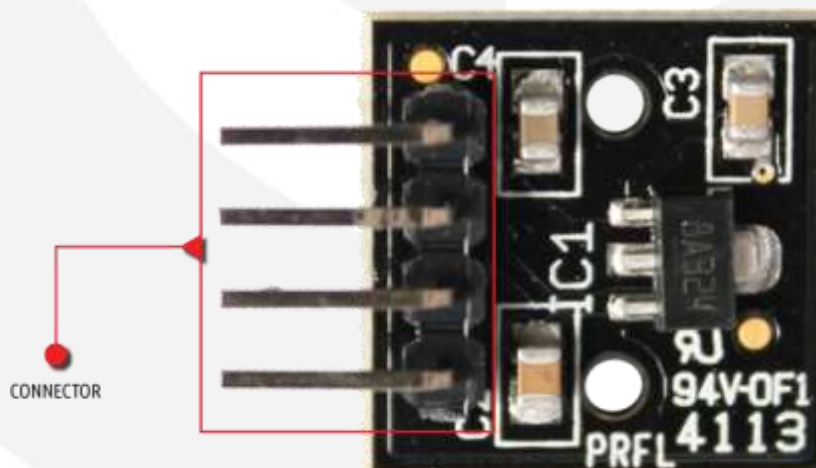


Figure 1-VCNL4000-BB bottom view

VCNL4000-BB has 4 pin male berg connector. The header has supply pins and I2C pins for interfacing with microcontroller. I2C slave address of VCNL4000 is 0x26, so read and write addresses are 0x26 and 0x27 respectively. Figure 2 shows pin layout for VCNL4000-BB.

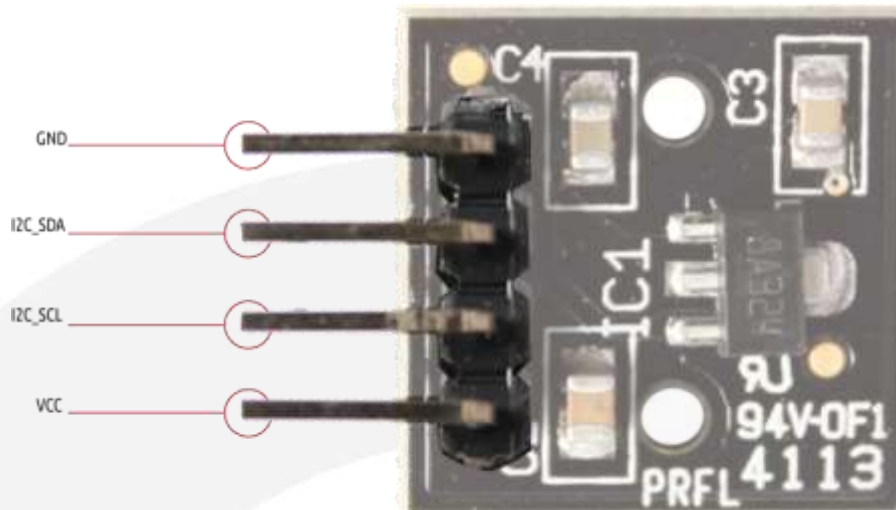


Figure 2-VCNL400-BB pin layout

## 5 Pseudo Code

### 5.1 Register Addresses and Bit Values for Flags

```
// VCNL4000 I2C slave address
VCNL_SLAVE_ADDR = 0x26
// VCNL4000 register addresses
VCNL_REG_CMD = 0x80
VCNL_REG_AMB_DATA_H = 0x85
VCNL_REG_AMB_DATA_L = 0x86
VCNL_REG_PROX_DATA_H = 0x87
VCNL_REG_PROX_DATA_L = 0x88
// VCNL4000 command register flag values
VCNL_START_AMBIENCE_CONVERSION = 0x10
VCNL_START_PROXIMITY_CONVERSION = 0x08
VCNL_AMBIENCE_DATA_READY = 0x40
VCNL_PROXIMITY_DATA_READY = 0x20
```





## 5.2 Reading VCNL4000 Register

```
uint8_t vcnlReadRegister(regAddr)
{
    uint8_t data;
    // I2C start signal
    i2cStart();
    // send VCNL4000 I2C slave address with R/W bit set as
    i2cWriteByte(VCNL_SLAVE_ADDR);
    // send register address to read
    i2cWriteByte(regAddr);
    // I2C repeated start signal
    i2cStart();
    // send VCNL4000 I2C address with R/W bit set as 1
    i2cWriteByte(VCNL_SLAVE_ADDR | 0x01);
    // read a byte from I2C
    data = i2cReadByte();
    // I2C stop signal
    i2cStop();
    return data;
}
```

## 5.3 Writing VCNL4000 Register

```
void vcnlWriteRegister(uint8_t regAddr, uint8_t data)
{
    // I2C start signal
    i2cStart();
    // send VCNL4000 I2C slave address with R/W bit set as 0
    i2cWriteByte(VCNL_SLAVE_ADDR);
    // send register address to write
    i2cWriteByte(regAddr);
    // write data byte
    i2cWriteByte(data);
    // I2C stop signal
    i2cStop();
}
```



## 5.4 Reading Ambient Light Value

```
uint16_t vcnlGetAmbience(void)
{
    uint16_t data;
    uint8_t temp, cmd;
    // read VCNL4000 command register
    // OR the new command with read data
    cmd = vcnlReadRegister(VCNL_REG_CMD);
    cmd |= VCNL_START_AMBIENCE_CONVERSION;
    // write new command to VCNL4000
    vcnlWriteRegister(VCNL_REG_CMD, cmd);
    // wait for proximity conversion to complete
    do
    {
        temp = vcnlReadRegister(VCNL_REG_CMD);
    }while( !(cmdReg & VCNL_AMBIENCE_DATA_READY) );
    // read ambience data high byte
    temp = vcnlReadRegister(VCNL_REG_AMB_DATA_H);
    data = temp << 8;
    // read ambience data low byte
    temp = vcnlReadRegister(VCNL_REG_AMB_DATA_L);
    data |= temp;
    return data;
}
```

## 5.5 Reading Proximity Distance Value

```
uint16_t vcnlGetProximity(void)
{
    uint16_t data;
    uint8_t temp, cmd;
    // read VCNL4000 command register
    // OR the new command with read data
    cmd = vcnlReadRegister(VCNL_REG_CMD);
    cmd |= VCNL_START_PROXIMITY_CONVERSION;
    // write new command to VCNL4000
    vcnlWriteRegister(VCNL_REG_CMD, cmd);
    // wait for min 400us after starting conversion
    // as mentioned in the datasheet
    delay_us(400);
    // wait for proximity conversion to complete
    do
    {
        temp = vcnlReadRegister(VCNL_REG_CMD);
    }while( !(cmdReg & VCNL_PROXIMITY_DATA_READY) );
    // read proximity data high byte
    temp = vcnlReadRegister(VCNL_REG_PROX_DATA_H);
    data = temp << 8;
    // read proximity data high byte
    temp = vcnlReadRegister(VCNL_REG_PROX_DATA_L);
    data |= temp;
    return data;
}
```



## 6 Reference

VCNL4000 Datasheet: <http://www.vishay.com/docs/83798/vcnl4000.pdf>

78L33 Datasheet: <http://in.mouser.com/Search/Refine.aspx?Keyword=78L33>





Phi Robotics Research Pvt. Ltd.

[www.phi-robotics.com](http://www.phi-robotics.com)