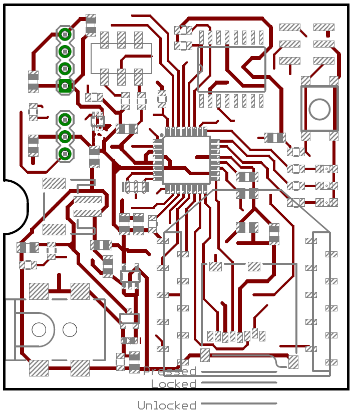
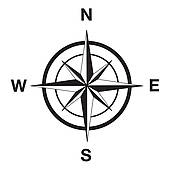
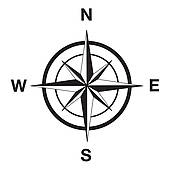
******Board Notes**

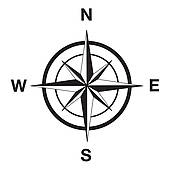
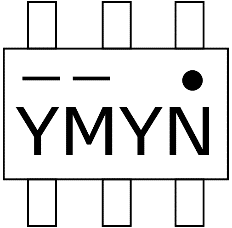
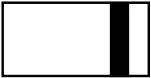
TOP

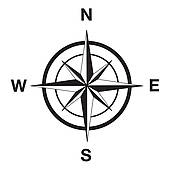
BOTTOM

**Arduino ISP Connector:**

|  |  |  |  |
| --- | --- | --- | --- |
| MISO |  | Top Right | VIN (3.3V) |
| SCK |  |  | GND |
| MOSI | Bot. Left |  | RESET |

**Parts Assembly:**

* Y1: Oscillator, orientation does not matter
* LEDs L1, L2, L3: Green marking towards East
* ****U3: IC markings look like this when looking closely:
* D1: Diode, marking towards East

**Sensor Module Connectors:**

|  |  |  |
| --- | --- | --- |
| GND |  |  |
| to SM’s RXI |  |  |
| to SM’s TXO |  |  |
| Controlled SM Supply Voltage |  |  |
|  |  |  |
| Continuous supply voltage from SM |  |  |
| Continuous supply voltage towards SM |  |  |
| Wakeup interrupt from SM |  |  |

DO NOT PLUG IN SUPPLY VOLTAGE JACK OF COMM. MODULE WHILE 3 PIN SENSOR MODULE CONNECTOR IS USED! (Because if sensor module is also externally powered, then there is a short between the two supply voltages through the “Continuous V from SM” or “Continuous V towards SM” pin)

For a periodic sensor module, only the 4 pin connector is needed. For an interrupting sensor module, the three pin connector is needed as well. Then the “Controlled SM Supply Voltage” pin does not have to be connected on the sensor module anymore, but it can be supplied through “Continuous supply voltage towards SM”, or if the sensor module has a constant power source, then the communication module can be supplied through “Continuous supply voltage from SM”.

Setup

* Program via Arduino ISP. In the Arduino IDE, the programmer that is selected must be “Arduino as ISP” and the program then upload using Sketch > Upload Using Programmer
* After assembling communication module: Burn bootloader via Arduino ISP (Tools > Burn Bootloader)
* If a XBee module is not found in XCTU, it is probably because it has been configured to be in sleep mode (SM = 1). To add the module in XCTU, you need to pull the sleep pin (= pin 9) to GND (= pin 10) while adding it and reading the parameters in XCTU.