# DESIGNING A SMART WEARABLE BAND

- **Purpose -** The purpose of this soft project is to develop a wireless heart rate monitoring device based on the principle of Photoplethysmography and using the Bluetooth Low Energy (BLE) technology. The emphasis here is mainly on the features used and to implement the firmware/RTOS development.

- **Introduction -** The development of wearable technologies and Bluetooth low energy (BLE) has brought various innovations in health care and monitoring. Earlier, certain health monitoring and measurement devices such as Electrocardiography (ECG) devices, pulse oximeters, and heart rate monitors, were only confined mainly in the intensive care units of hospitals and clinics. These devices are usually complex, expensive and largely non-portable. With the advent of wearable's and BLE, monitoring of the body could be done anywhere now.

- **Aim -** The aim of this soft project is to design a circuit diagram using given features like heart rate sensor (Photoplethysmogram sensor), Inertial measurement unit, a low powered MCU and a temperature sensor and further to comment or discuss the firmware or RTOS development.

- **Principles and Technologies –** This project involves two principles and technologies used. One is **Photoplethysmography** and the other is **Bluetooth Low Energy.**

- **Materials -** The main materials used in this project include nRF51822 system-on-chip (SoC) from Nordic Semiconductors, which serves both as the controller and BLE communication unit, MAX30100 i.e. Pulse oximeter, which served as the sensor unit. These will be briefly discussed here along the line of what they are, what they offer and their suitability for the project.

  - **nRF51822 -** nRF51822 belongs to the nRF51 Series from Nordic Semiconductors, which is a family of multi-protocol SoC devices for wireless applications distinguished by their high flexibility and ultra-low power consumption. It is designed ideally for BLE and ultra-low-power wireless applications. It also has a built-in set of analog and digital peripherals that can work together through what is known as the Programmable Peripheral Interconnect (PPI) system without the intervention of CPU. Figure given below is a schematic of nRF51822 chip showing both the general purpose input-output pins and special purpose pins with their numbers and names.

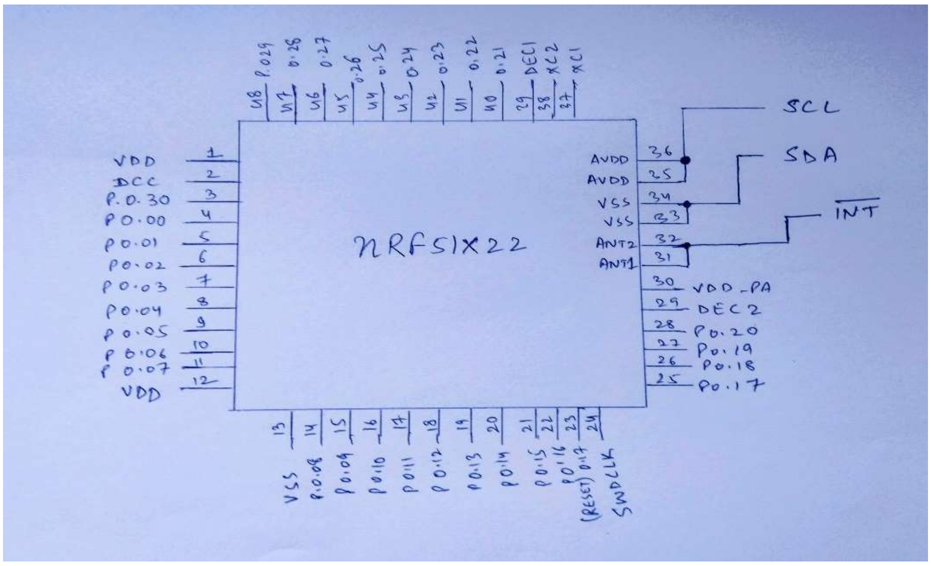


**Fig. 1 Schematic of nRF51822 chip showing its pin outs.**

➢ **MAX30100 -** MAX30100 is a pulse oximeter and heart-rate sensor integrated circuit (IC) for wearable health monitoring systems or devices. It detects pulse oximetry and heart rate signals with a combination of two LEDs (red and infra-red), a photo detector, optimized optics, and low-noise analog signal processing techniques. It can operate from either 1.8V or 3.3V power supplies and can be powered down programmatically by software with negligible standby current, thereby presenting the possibility of leaving the power supply connected all the time. Figure given below shows the pin configuration of MAX30100.

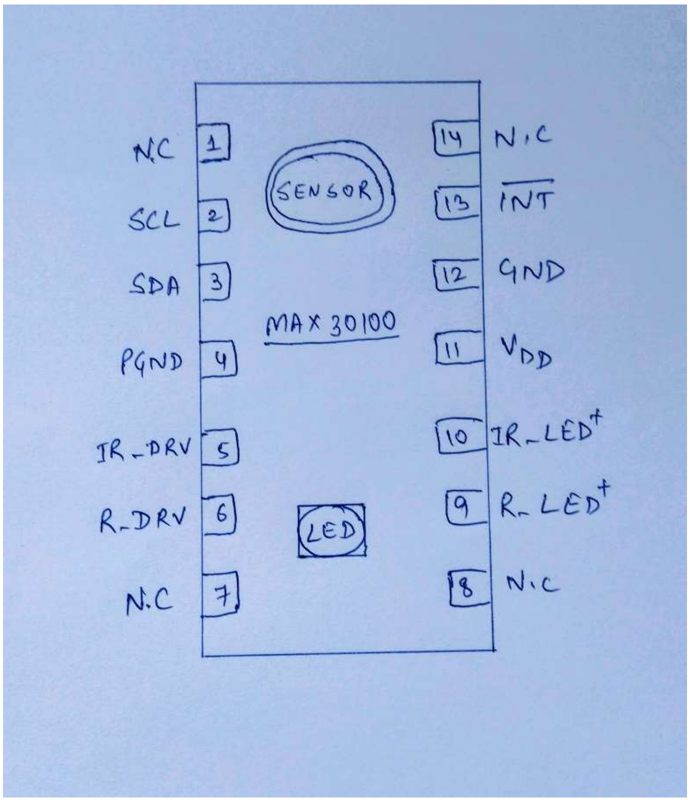


**Fig. 2 Pin Configuration Diagram of MAX30100**

- **Methods –** Two methods were involved to achieve the desired results. They are -

➢ **Sensor Unit -** The development of the sensor unit was done in three major steps. The first step was the preparation of the hardware setup, which included designing and building a break out sensor board for MAX30100 and connecting the appropriate pins from nRF51 evaluation board to MAX30100 board. The second step was to setup the development environment for firmware development. The third step was the actual implementation of the firmware.

➢ **Hardware setup for the Sensor -** The heart-rate sensor IC used in this project, max30100, comes in a very small form (5.6mm x 2.8mm x 1.2mm) and therefore needs at least a break out sensor board to make it usable for development process. It also requires some pull-up resistors and capacitors to work properly. So the first task was to make a breakout sensor board.

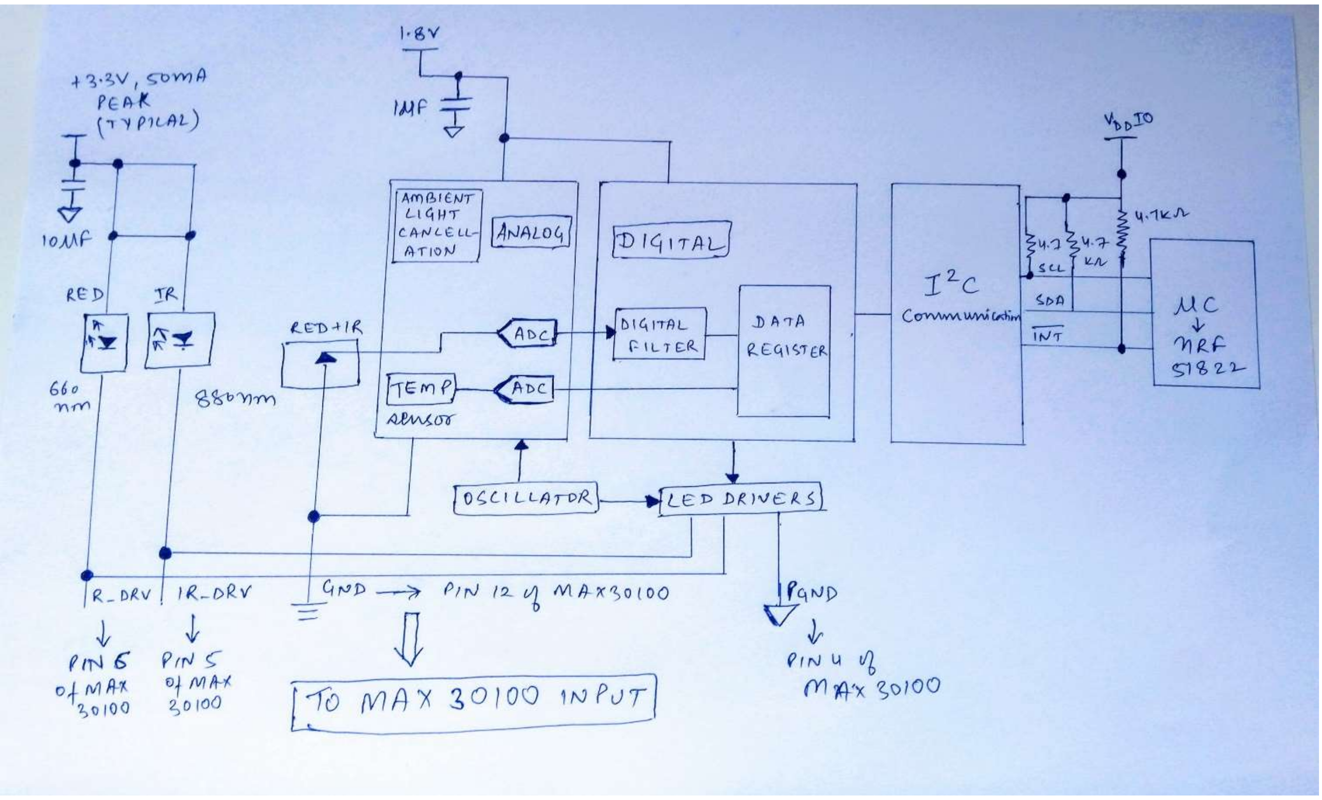


**Fig. 3 MAX30100 Schematic of application circuit**

- The figures 1, 2 and 3 shows the schematic of a complete application circuit connecting nRF BLE with MAX 30100. The data is process from nRF to the break out sensor board circuit consisting of I2C communication protocol as well as Digital and Analog sections. The Analog section consists of temperature sensing IC board after the conversion of digital data into analog one by two ADC circuits. Further to this, the break out sensor board circuit is connected to the MAX30100 by R_DRV, IR_DRV, GND and PGND pins as shown in Fig. 3

- **Firmware Development Environment Setup -** The first step in the firmware development was to set up the development environment. Linux OS was used for the same. The tools required were gcc cross compiler for ARM processors, required to compile the source code written for nRF51822 on a Linux machine.

- **Firmware Implementation -** The basic logic of the firmware is found in the main routine :

  - **Main function of the firmware -**

```
int main (void)
{
        twi_master_init(); init _sensor();
        ble_stack_init(); timer_init();
        gpio_init(); gap_params_init();
        services_init(); advertising_init();
        conn_params_init(); sec_params_init();
        advertising_start();
```

```
        while (true)
        {
        power_manage();
        }
}
```

- **Initialize I2C master –**

```
bool twi_master_init(void)
{
        TWI_SDA_STANDARD0_NODRIVE1();
        TWI_SCL_STANDARD0_NODRIVE1();
        TWI_SCL_HIGH(); TWI_SCL_OUTPUT();
        TWI_SDA_HIGH(); TWI_SDA_OUTPUT();
        return twi_master_clear_bus();
}
```

- **Initialize sensor function –**

```
static void init_sensor()
{
        config.spo2_config = (SPO2_HI_RES_EN | SPO2_SR_100);
        config.mode_config = HR_ONLY_EN;
        config.led_config =  (RED_PA_50_0 | IR_PA_50_0);
        initial_config(config);
}
```

- **Initialize timers function –**

```
static void timers_init(void)
{
        APP_TIMER_INIT(APP_TIMER_PRESCALER, APP_TIMER_MAX_TIMERS,
        APP_TIMER_OP_QUEUE_SIZE, false);
}
```

- **Initializing GPIO pins –**

```
static void gpio_init(void)
{
        simple_uart_config(RTS_PIN_NUMBER, TX_PIN_NUMBER,
                CTS_PIN_NUMBER,  RX_PIN_NUMBER, HWFC);
        NRF_GPIO->PIN_CNF[INTERRUPT_PIN]  =
                (GPIO_PIN_CNF_SENSE_Low << GPIO_PIN_CNF_SENSE_Pos)
                |(GPIO_PIN_CNF_DRIVE_S0S1 << GPIO_PIN_CNF_DRIVE_Pos)
                |(NRF_GPIO_PIN_NOPULL << GPIO_PIN_CNF_PULL_Pos)
                |(GPIO_PIN_CNF_INPUT_Connect << GPIO_PIN_CNF_INPUT_Pos)
                |(GPIO_PIN_CNF_DIR_Input << GPIO_PIN_CNF_DIR_Pos);

        NVIC_EnableIRQ(GPIOTE_IRQn);
```

```
        NRF_GPIOTE->INTENSET = GPIOTE_INTENSET_PORT_Set <<
                GPIOTE_INTENSET_PORT_Pos;
        sd_nvic_SetPriority(GPIOTE_IRQn, NRF_APP_PRIORITY_LOW);
}
```

- **Sensor security parameters initialization function –**

```
static void sec_params_init(void)
{
        m_sec_params.timeout          =    SEC_PARAM_TIMEOUT;
        m_sec_params.bond             =    SEC_PARAM_BOND;
        m_sec_params.mitm             =    SEC_PARAM_MITM;
        m_sec_params.io_caps          =    SEC_PARAM_IO_CAPABILITIES;
        m_sec_params.oob              =    SEC_PARAM_OOB;
        m_sec_params.min_key_size     =    SEC_PARAM_MIN_KEY_SIZE;
        m_sec_params.max_key_size     =    SEC_PARAM_MAX_KEY_SIZE;
}
```

- **Heart Rate Calculation -** In PPG-based heart rate measurement, heart rate is determined by the number of peaks detected in the PPG signal per minute. Thus each peak in the signal represents one heart beat. The application calculates the heart rate for every 1000 samples. Since the sampling rate of the sensor is set at 100 samples per second, the application calculates the heart rate every 10 seconds.

```
private void calculateHR(final ArrayList<Integer> samples){
 for(int i = 0; i < samples.size(); i++){
  for(int i = 0; i < samples.size(); i++) {
    double cur = samples.get(i);
     if (cur > mx) {
        mx = cur; mxPos = i;
     }
     if (cur < mn) {
        mn = cur; mnPos = I;
     }
     if(lookForMax == 1){
        if(cur < (mx - delta )){
                mMaxPoints.add(mx);
                mMaxPos.add(mxPos);
                mn = cur; mnPos = i; lookForMax = 0;
     }
  } else {
        if(cur > (mn + delta)){
```

```
                mMinPoints.add(mn);
                mMinPos.add(mnPos);
                mx = cur; mxPos = i; lookForMax = 1;
            }
        }
    }
    double peaks = mMaxPoints.size();
    double duration = samples.size()/SAMPLING_RATE;
    mHr = (int)Math.round((peaks*60)/duration);
  }
}
```

- **Conclusion -** The main goal was to work with MAX30100 for sensing PPG (Photoplethysmogram) signal. For this I have used nRF5822 for sensor-side control and BLE communication. The emphasis in this project was working mostly on the circuit diagrams and the basic functionality of the complete set-up.