ELECTRO MECHANICAL CPR

PROJECT PRELIMINARY REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF DEGREE OF

BACHELOR OF TECHNOLOGY

in

ELECTRICAL AND ELECTRONICS ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

by

NIKHIL M B (VAS19EE040)
PRINCE SURESH V S (VAS19EE041)
YEDHUKRISHNA K S (VAS19EE051)
KRISHNAPRASAD K A (LVAS19EE064)

(AN ISO 9001:2015 CERTIFIED INSTITUTION)

Department of Electrical and Electronics Engineering

Vidya Academy of Science & Technology
Thalakkottukara, Thrissur - 680 501
(http://www.vidyaacademy.ac.in)

January 2023

Department of Electrical and Electronics Engineering
Vidya Academy of Science & Technology

Thalakkottukara, Thrissur - 680 501

(http://www.vidyaacademy.ac.in)

(AN ISO 9001:2015 CERTIFIED INSTITUTION)

Certificate

This is to certify that the Preliminary Project Report titled **"ELECTRO MECHANICAL CPR"** is a bonafide record of the work carried out by **NIKHIL M B, PRINCE SURESH V S, YEDHUKRISHNA K S, KRISHNAPRASAD K A (VAS19EE040, VAS19EE041, VAS19EE051, LVAS19EE064)** of Vidya Academy of Science & Technology, Thalakkottukara, Thrissur - 680 501 in partial fulfillment of the requirements for the award of **Degree of Bachelor of Technology** in **Electrical and Electronics Engineering** of **APJ Abdul Kalam Technological University**, during the academic year 2019-2023. The Preliminary Project report has been approved as it satisfies the academic requirements in the respect of preliminary project work prescribed for the said degree.

Project Guide	**Head of Department**
ANOOJA V S	Dr Mary P. Varghese
Asst. Prof., Dept. of EEE	Asso., Dept. of EEE

Project Co-guide

ATHIRA P S

Asst. Prof., Dept. of EEE

Undertaking

We, **NIKHIL M B, PRINCE SURESH V S, YEDHUKRISHNA K S, KRISHNAPRASAD K A (Univ. Reg. No. VAS19EE040, VAS19EE041, VAS19EE051, LVAS19EE064)**, hereby undertake that the main project work entitled **"ELECTRO MECHANICAL CPR"**, is carried out by us independently under the valuable guidance of **ANOOJA V S**, Asst. Prof., and **ATHIRA P S**, Asst. Prof., Dept. of Electrical and Electronics Engineering, Vidya Academy of Science and Technology, Thalakkottukara, Thrissur, in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology in Electrical and Electronics Engineering** of **APJ Abdul Kalam Technological University**, during the academic year 2019-2023.

Thrissur **NIKHIL M B, PRINCE SURESH V S, YEDHUKRISHNA K S,**
January 2023 **KRISHNAPRASAD K A**

Acknowledgement

During the course of our main project work several persons collaborated directly and indirectly with me. Without their support it would be impossible for me to finish our work. That is why we wish to dedicate this section to recognize their support.

We want to start expressing our thanks to our project guide, **ANOOJA V S**, Asst. Prof., Dept. of EEE & project Co-guide, **ATHIRA P S**, Asst. Prof., Dept. of EEE, because of his valuable advice and guidance towards this work. We received motivation, encouragement and hold up from them during the course of work.

We would like to express our sincere gratitude to project coordinators **Ms Neenu Thomas**, Asst. Prof. Dept. of EEE and **Mr Arun Xavier**. Asst. Prof., of EEE department for their valuable suggestion and feedbacks.

We are grateful to express our thanks to all the faculty members of our department for their support. We articulate our gratitude to all our friends for their support and help for this work.

We are thankful to **Dr Mary P. Varghese**, Head of Electrical and Electronics Engineering Department, and our Principal **Dr Saji C. B.**, for their sole co-operation.

Last, but not the least We wish to express our gratitude to God Almighty for His abundant blessings without which this effort would not have been successful.

NIKHIL M B, PRINCE SURESH V S, YEDHUKRISHNA K S, KRISHNAPRASAD K A
Reg. No. VAS19EE040, VAS19EE041, VAS19EE051, LVAS19EE064
Seventh Semester B.Tech (2019 Admissions)
Vidya Academy of Science & Technology
Thrissur - 680 501.

January 2023

Abstract

An electro-mechanical CPR device utilizing Arduino technology and a suction cup attachment has been developed to provide both chest compressions and ventilation to a person in cardiac arrest. The device consists of a small electric motor , which is attached to the suction cup and provides up and down movements to deliver compressions to the chest. A ventilation system is also included, which provides breaths to the patient through a facemask or endotracheal tube. The device is activated by pressing a button and can be programmed using Arduino technology to deliver compressions at a specific rate and depth. The suction cup attachment allows the device to be easily secured to the patient's chest, ensuring proper placement and consistent compressions. The motor and suction cup arrangement allows for efficient delivery of compressions without the need for manual intervention.The device is designed to be used by first responders or laypeople in emergency situations. It is intended to provide a more efficient and effective form of CPR compared to manual compressions, which can be tiring for rescuers and may not be performed with enough force or at the correct rate.

Contents

CERTIFICATE

UNDERTAKING

ACKNOWLEDGEMENT **i**

ABSTRACT **ii**

LIST OF FIGURES **iv**

LIST OF TABLES **v**

LIST OF SYMBOLS AND ABBREVIATIONS **v**

1 INTRODUCTION **1**

1.1 Objectives of the Work 1

1.2 Motivation for this work 2

1.3 Methodologies Adopted 2

1.4 Outline of the Report 3

1.5 Summary 4

2 LITERATURE REVIEW **5**

2.1 Mechanical Chest Compression Devices: Historical Evolution, Classification and Current Practices, A Short Review [1] 5

2.2 Design of a Low-cost Automated Cardiopulmonary Resuscitation Device with Piston-Driven Chest Compression System[2] 6

2.3 Simulation of an Electro-Mechanical Resuscitation Device for Cardiopulmonary Resuscitation[3] 7

2.4 Summary 8

3 METHODOLOGIES FOR THE PROJECT 9

3.1 Block Diagram 9

3.2 Circuit Diagram 10

3.3 Positioning the device 11

3.4 Begin compression and ventilation 11

3.5 Monitor the person 11

3.6 Continue compressions 12

3.7 Summary 12

4 HARDWARE SETUP OF THE PROJECT 13

4.1 Introduction 13

4.2 Hardware Setup of the Project 13

4.3 Hardware Components 14

4.3.1 Arduino Board 14

4.3.2 Pulse and Blood oxygen sensor 14

4.3.3 LCD Display 15

4.3.4 Jumper Wire 16

4.4 Summary 16

5 CONCLUSION AND SCOPE OF FUTURE WORK 18

BIBLIOGRAPHY 19

APPENDIX 20

List of Figures

1.1 Basic Model 3

3.1 Block Diagram 9

3.2 Circuit Diagram 10

4.1 Full Hardware Setup 13

4.2 Arduino Chip 14

4.3 MAX 3010 15

4.4 LCD Display 15

4.5 Jumper Wire 16

4.6 Initially 16

4.7 Final Result 17

List of Symbols and Abbreviations

CPR	Cardiopulmonary Resuscitation
mCPR	Mechanical Cardiopulmonary Resuscitation
MCCD	Mechanical chest compression device
PWM	Pulse-width modulation
RPM	Revolutions per minute
DC	Direct current
LCD	Liquid-crystal display
MHz	Megahertz
VCC	Common Collector Voltage
GND	Ground
SDA	Serial Data
SCL	Serial Clock
LED	Light-emitting diode
SpO2	Saturation of Peripheral Oxygen
BPM	Beats per minute
ICSP	In-circuit serial programming

Chapter 1

INTRODUCTION

CPR is a lifesaving technique that's useful in many emergencies, such as a heart attack, in which someone's breathing or heartbeat has stopped. With manual CPR, many factors come into play, including fatigue, physical abilities, focus on several tasks, fear of doing harm to the victim, and complexity of the CPR procedure. This project deals with every problem of manual CPR and gives a cheap and secure method of tackling problems with efficiency. mCPR may be particularly useful in situations where traditional CPR is difficult to perform, and has been shown to be effective in both in-hospital and out-of-hospital cardiac arrest situations. The additional ventilation mechanism provided enables stable and sufficient amount of oxygen at specified intervals and thus increases the chance of survival.

1.1 Objectives of the Work

The objectives of the electro-mechanical CPR device project are as follows:

- To develop a portable, handheld device that can deliver consistent and properly administered chest compressions and ventilation to a person in cardiac arrest.
- To incorporate pulse monitoring capabilities into the device to provide real-time feedback on the patient's vital signs.
- To utilize Arduino technology to allow for customization and flexibility in programming the device to deliver compressions at a specific rate and depth.

- To design a device that is cost-effective and accessible to a wider range of healthcare professionals and first responders.
- To improve patient outcomes and increase the chances of survival for individuals experiencing cardiac arrest through the use of the electro-mechanical CPR device.
- To evaluate the effectiveness and usability of the device through clinical trials and user feedback.
- To make the device available at a much more affordable price and thus enable it to reach a wide range of people

1.2 Motivation for this work

The motivation for the electro-mechanical CPR device project was to develop a more affordable and comprehensive alternative to existing CPR devices on the market. The current devices available can be expensive, with some costing more than 10 lakh rupees and still not provide a proper ventilation system in addition to chest compressions or have pulse monitoring capabilities. This motivated our team to design a device that was more cost-effective, provided both chest compressions and ventilation, and had pulse monitoring capabilities. The use of an electro-mechanical device allows for consistent and properly administered compressions without the need for manual intervention, and the inclusion of a ventilation system and pulse monitoring feature increases the device's effectiveness in resuscitating patients. The goal was to create a device that was accessible to a wider range of healthcare professionals and first responders, and that had the potential to improve patient outcomes and save lives.

1.3 Methodologies Adopted

The basic methodology for using mechanical CPR devices involves the following steps:

Position the device: Place the device on the person's chest, aligning it with the lower half of the sternum.

Start the device: Turn on the device and set it to the appropriate compression rate and depth.

Begin compression and ventilation: The device will deliver chest compressions automatically along with ventilation at specific intervals of time.

Monitor the person: Keep an eye on the person's condition and be prepared to stop the compressions if they show signs of waking up or if their condition deteriorates.

Continue compressions: Continue delivering chest compressions and ventilation until the pulse goes back to normal.

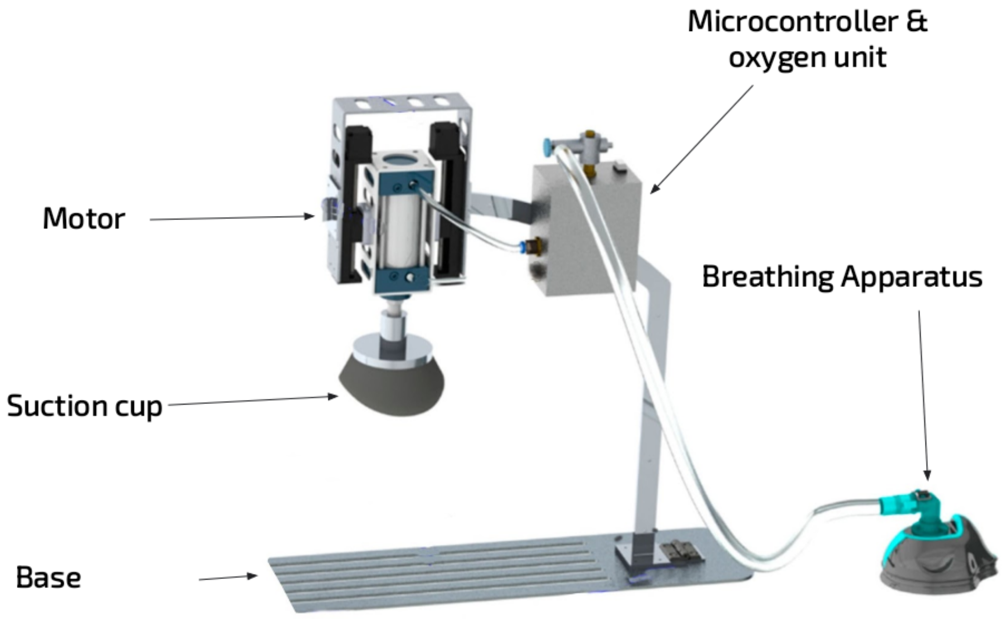


Figure 1.1: Basic Model

1.4 Outline of the Report

This report contains 5 chapters

- Chapter 1: Contains the introduction to the project work and describes the objective and motivation behind the work.
- Chapter 2: Contains literature review based on 3 reference papers specifying the developments in the proposed system.
- Chapter 3: Specifies the general methodology along with diagrams and various steps involved to do mCPR properly.

- Chapter 4: Contains hardware setup of the project
- Chapter 5: Concludes the report and mentions the scope of future works for mechanical CPR.

1.5 Summary

This chapter provides a brief introduction to mechanical CPR and the various objectives behind the proposed system.It specifies all the advantages it has to a conventional CPR.Methodology that specifies the basics steps to implement mCPR and how it can make a difference in the survival of a person is also mentioned.This project deals with every problem of manual CPR and gives a cheap and secure method of tackling problems with efficiency.The additional ventilation mechanism provided enables stable and sufficient amount of oxygen at specified intervals and thus increases the chance of survival.

Chapter 2

LITERATURE REVIEW

Literature review of electromechanical CPR devices refers to the systematic examination and evaluation of research and scholarly work that has been published on the topic of electromechanical devices for cardiopulmonary resuscitation (CPR). These devices use an electric motor to assist in the delivery of chest compressions during CPR, with the aim of improving the effectiveness and/or ease of CPR compared to manual methods. Literature review of electromechanical CPR devices typically includes a review of the history and development of these devices, their current state of use and effectiveness, and any ongoing research or challenges in their development and implementation. Such literature reviews can provide valuable insights into the potential benefits and limitations of electromechanical CPR devices and inform the development of guidelines and best practices for their use in clinical settings.

2.1 Mechanical Chest Compression Devices: Historical Evolution, Classification and Current Practices, A Short Review [1]

The paper ”Mechanical Chest Compression Devices: Historical Evolution, Classification and Current Practices, A Short Review” provides an overview of the development and current use of mechanical devices for chest compressions in the field of cardiopulmonary resuscitation (CPR).

The introduction of the paper notes that CPR has been a critical intervention for the management of cardiac arrest for over 50 years. Manual CPR has been the mainstay of treatment, but there are limitations to its effectiveness, including the need for skilled personnel and the potential for rescuer fatigue. In response to these limitations, mechanical devices for chest compressions have been developed as an alternative or supplement to manual CPR.

The paper goes on to classify mechanical chest compression devices into three categories: manual, semi-automatic, and automatic. Manual devices require the operator to manually compress the chest, while semi-automatic devices use a motor to assist in the compressions. Automatic devices are fully self-contained and do not require operator input for compressions.

The paper then discusses the various mechanical devices that have been developed over the years and their relative advantages and disadvantages. It also discusses current practices and guidelines for the use of these devices in clinical settings.

In conclusion, the paper highlights the importance of mechanical chest compression devices as an alternative to manual CPR and notes that they have the potential to improve the effectiveness of CPR, particularly in situations where skilled personnel are not available or when rescuer fatigue is a concern. However, the authors caution that further research is needed to fully understand the optimal use and potential benefits of these devices in clinical practice.

2.2 Design of a Low-cost Automated Cardiopulmonary Resuscitation Device with Piston-Driven Chest Compression System[2]

The paper ”Design of a Low-cost Automated Cardiopulmonary Resuscitation Device with Piston-Driven Chest Compression System” presents the design and development of a low-cost, automated CPR device.

The introduction of the paper notes that CPR is a critical intervention for the management of cardiac arrest, but manual CPR has limitations, including the need for skilled personnel and the potential for rescuer fatigue. Mechanical devices for chest compressions have been developed as an alternative or supplement to manual CPR, but many of these devices are expensive and not widely available in resource-limited settings.

In response to this need, the authors set out to design and develop a low-cost, automated CPR device that could be used in resource-limited settings. The device utilizes a piston-driven chest compression system and is intended to be easy to use, portable, and affordable.

The paper goes on to describe the design and development of the device, including the selection of materials and components, the development of a prototype, and the testing and evaluation of the prototype.

In conclusion, the paper presents the successful development of a low-cost, automated CPR device with a piston-driven chest compression system. The authors note that the device has the potential to improve the availability and effectiveness of CPR in resource-limited settings and could be a valuable addition to the range of mechanical CPR devices available.

2.3 Simulation of an Electro-Mechanical Resuscitation Device for Cardiopulmonary Resuscitation[3]

The paper ”Simulation of an Electro-Mechanical Resuscitation Device for Cardiopulmonary Resuscitation” presents the simulation of an electro-mechanical resuscitation (EMR) device for CPR.

The introduction of the paper notes that CPR is a critical intervention for the management of cardiac arrest, but manual CPR has limitations, including the need for skilled personnel and the potential for rescuer fatigue. Mechanical devices for chest compressions have been developed as an alternative or supplement to manual CPR, but many of these devices are expensive and not widely available.

In response to this need, the authors set out to design and simulate an EMR device for CPR that could be used in a variety of settings. The device utilizes an electric motor to drive a piston that delivers chest compressions.

The paper goes on to describe the simulation of the EMR device, including the selection of materials and components, the development of a prototype, and the testing and evaluation of the prototype.

In conclusion, the paper presents the successful simulation of an EMR device for

CPR. The authors note that the device has the potential to improve the availability and effectiveness of CPR in a variety of settings and could be a valuable addition to the range of mechanical CPR devices available.

2.4 Summary

A literature review of mechanical chest compression devices (MCCDs) may discuss the historical evolution and classification of these devices, as well as their current practices in cardiopulmonary resuscitation (CPR). The review may also examine the potential advantages and limitations of MCCDs compared to manual CPR.

The two specific articles that is mentioned, "Design of a Low-cost Automated Cardiopulmonary Resuscitation Device with Piston-Driven Chest Compression System" and "Simulation of an Electro-Mechanical Resuscitation Device for Cardiopulmonary Resuscitation," both focus on the development and evaluation of MCCDs. The first article presents the design of a low-cost MCCD using a piston-driven system, while the second article conducts a simulation study of an electro-mechanical resuscitation device for CPR.

In terms of common features, both articles aim to improve upon existing MCCD designs and to enhance the effectiveness of CPR through the use of mechanical devices. However, there is room for future improvements in MCCDs, such as further optimizing their mechanical principles and ensuring their reliability and durability. Additionally, further research may be needed to determine the most appropriate circumstances for the use of MCCDs and to assess their clinical effectiveness in real-world settings.

Chapter 3

METHODOLOGIES FOR THE PROJECT

3.1 Block Diagram

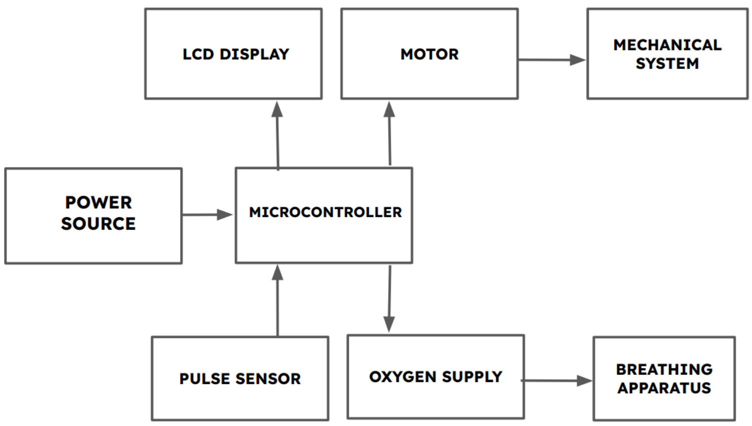


Figure 3.1: Block Diagram

- Power supply: 12v 7Ah battery is used as the power source in this project
- Microcontroller: Arduino UNO R3 is chosen as the microcontroller for controlling the entire system. Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button.
- Pulse sensor: The MAX30100 pulse oximeter and heart rate sensor is an I2C-based low-power plug-and-play biometric sensor. It is used to sense the pulse

for monitoring and for controlling the CPR mechanism. The module features the MAX30100 – a modern, integrated pulse oximeter and heart rate sensor IC, from Analog Devices. It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry (SpO2) and heart rate (HR) signals.

- Motor : A 12v 150RPM DC geared motor which can provide high torque is used in this project for providing chest compressions.
- Mechanical system: A mechanical system with reciprocating mechanism is designed to give compression as per required.
- Oxygen supply and Breathing apparatus: Oxygen supply unit provides oxygen through the attachable breathing apparatus when ever it is triggered by the micro-controller.

3.2 Circuit Diagram

This is a model of how the connection look like :

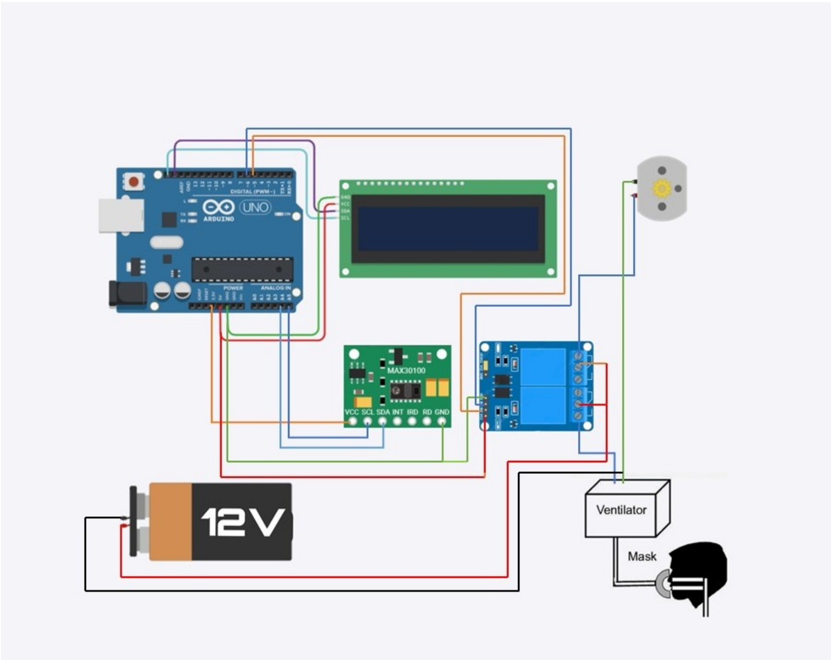


Figure 3.2: Circuit Diagram

3.3 Positioning the device

To begin the electromechanical CPR procedure, you should first position the device on the patient's chest, aligning it with the lower half of the sternum. This is an important step in ensuring that the device is properly positioned to deliver effective chest compressions. The device should be placed on the patient's chest so that it is centered over the sternum, which is the long, flat bone that runs down the middle of the chest. By aligning the device with the lower half of the sternum, you can ensure that the chest compressions are being delivered to the right part of the patient's body. It is important to take the time to properly position the device, as this will help to ensure that the procedure is as effective as possible.

3.4 Begin compression and ventilation

Once the electromechanical CPR device is properly positioned on the patient's chest, the next step is to start the device. This involves turning on the device and setting it to the appropriate compression rate and depth. The compression rate refers to the number of compressions that the device delivers per minute, and the depth refers to how far the device pushes down on the patient's chest during each compression. It is important to set the device to the appropriate settings, as this will help to ensure that the chest compressions are effective and that the patient is receiving the correct amount of CPR.

3.5 Monitor the person

As you perform electromechanical CPR on a patient, it is important to closely monitor their condition and be prepared to stop the compressions if needed. This involves keeping an eye on the patient's vital signs and being alert for any changes in their condition. If the patient shows signs of waking up or if their condition deteriorates, you should be prepared to stop the compressions and take appropriate action. For example, if the patient begins to wake up, you may need to stop the compressions and assist them in breathing on their own. If the patient's condition deteriorates, you may need to stop the compressions and consider administering advanced life support measures, such as defibrillation or medications. By closely monitoring the patient's condition, you can ensure that the procedure is being administered safely and effectively.

3.6 Continue compressions

Once you have started the electromechanical CPR procedure and are delivering chest compressions to the patient, it is important to continue delivering these compressions until advanced life support measures can be administered or until the patient regains a pulse. Chest compressions are an essential part of CPR, and they help to restore blood circulation to the body by manually pumping blood through the heart and into the arteries. By continuing to deliver chest compressions, you can help to keep the patient's vital organs, such as the brain and heart, supplied with oxygen-rich blood. You should continue delivering chest compressions until advanced life support measures, such as defibrillation or medications, can be administered. Alternatively, you should continue compressions until the patient regains a pulse, which indicates that the heart is beating on its own and is able to pump blood effectively.

3.7 Summary

Electromechanical CPR is a variation of CPR that involves the use of a mechanical device to assist with chest compressions. The goal of is to provide more consistent and efficient chest compressions than manual CPR. To perform CPR, you should first position the device on the patient's chest, aligning it with the lower half of the sternum. Next, turn on the device and set it to the appropriate compression rate and depth. The device will then deliver chest compressions and, in some cases, corresponding ventilations automatically. As you perform, it is important to monitor the patient's condition and be prepared to stop the compressions if needed. You should continue delivering chest compressions until advanced life support measures, such as defibrillation or medications, can be administered or until the patient regains a pulse.

Chapter 4

HARDWARE SETUP OF THE PROJECT

4.1 Introduction

The Arduino microcontroller will be the brain of the device, controlling all other hardware components and running the program logic. By using the Max30100 pulse sensor detection of user's heartrate and pulse oxymetry is done. All this is being displayed on the I2C LCD display. All these devices are powered by a 12v battery source by using necessary power conversions.

4.2 Hardware Setup of the Project

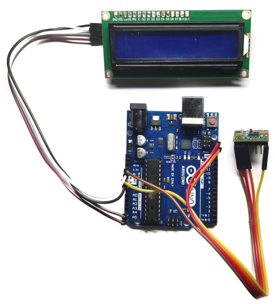


Figure 4.1: Full Hardware Setup

The MAX30100 has four pins: VCC, GND, SDA and SCL. VCC is connected to the 3.3V of Arduino, GND to GND of Arduino, SDA to A4 of Arduino and SCL to A5 of Arduino. The I2C LCD display has four pins: VCC, GND, SDA and SCL. VCC is connected to 5V of Arduino, GND to GND of Arduino, SDA to SDA of Arduino and SCL to SCL of Arduino. Then Arduino is powered by a DC source.

4.3 Hardware Components

4.3.1 Arduino Board

Figure 4.2: Arduino Chip

Arduino Uno is a microcontroller board based on the ATmega328 microcontroller. It has 14 digital input/output pins, 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It is an open-source platform that is widely used for building electronics projects because of its simplicity and flexibility. The Arduino Uno can be programmed using the Arduino Integrated Development Environment (IDE) and can be used to control a wide variety of sensors, actuators, and other devices. The R3 version of the Arduino Uno is the third revision of the board, which has improved connectivity with additional connectors and improved power management.

4.3.2 Pulse and Blood oxygen sensor

The MAX30100 is an integrated pulse oximetry and heart rate monitor sensor. It is commonly used to measure the blood oxygen saturation level (SpO2) and heart rate of a person. The sensor uses a red LED and an infrared LED along with a photodetector to measure the absorption of light by the blood.

To connect the MAX30100 to an Arduino board, follow these steps:

Connect the VCC pin of the sensor to the 3.3V pin of the Arduino. Connect the GND pin of the sensor to the GND pin of the Arduino. Connect the SDA pin of the sensor to the A4 pin of the Arduino. Connect the SCL pin of the sensor to the A5 pin of the

Figure 4.3: MAX 3010

Arduino. Once the hardware connection is established, we can use the Arduino IDE to write a program to read data from the MAX30100 sensor.We can also use libraries such as the MAX30100 library to make it easier to read data from the sensor.

4.3.3 LCD Display

Figure 4.4: LCD Display

An I2C LCD is a type of liquid crystal display (LCD) that uses the I2C communication protocol to control the display. It has a small number of pins, making it easy to connect to microcontrollers such as an Arduino. The I2C LCD uses a simple two-wire interface to communicate with the microcontroller, which reduces the number of pins required to connect the LCD to the microcontroller.

To connect an I2C LCD to an Arduino board, follow these steps:

Connect the GND pin of the LCD to the GND pin of the Arduino. Connect the VCC pin of the LCD to the 5V pin of the Arduino. Connect the SDA pin of the LCD to the A4 pin of the Arduino. Connect the SCL pin of the LCD to the A5 pin of the Arduino. Once the hardware connection is established, we can use the Arduino IDE to write a program to control the LCD. We can also use libraries such as the Liquid Crystal_I2C library to make it easier to control the LCD.

4.3.4 Jumper Wire

Jumper wires are short wires with connectors on either end, which are used to connect electronic components. Jumper wires come in various lengths and colors, and they are usually made of stranded copper wire covered with insulation. To use jumper wires with an Arduino board, simply need to connect the connectors on the ends of the jumper wires to the appropriate pins on the Arduino board and on the electronic components are using It is important to ensure that the jumper wire is properly connected to the correct pins, as incorrect connections can cause damage to the components or the Arduino board.

Figure 4.5: Jumper Wire

4.4 Summary

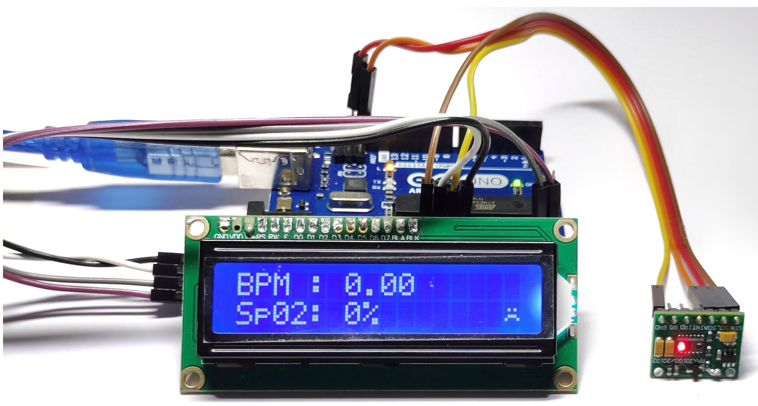


Figure 4.6: Initially

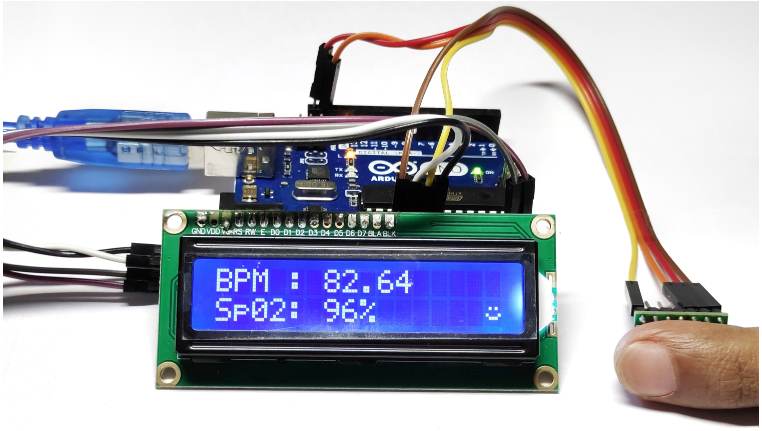


Figure 4.7: Final Result

Here as shown in the Figure 4.7 there exist the complete setup consisting of lcd display,pulse sensor and arduino board.The lcd screen shows 0 BPM(Beats Per Minute) and 0% SpO2(Saturation of Peripheral Oxygen) initially.But when the finger is placed over the sensor the pulse sensor measures the BPM and SpO2 levels accurately,this is shown in Figure 4.8.Hence the desired results were obtained..

Chapter 5

CONCLUSION AND SCOPE OF FUTURE WORK

Mechanical CPR has the advantage of providing consistent and continuous chest compressions, which can improve the chances of survival for a person in cardiac arrest. It can also reduce the fatigue of the rescuer, as the device does the majority of the work.

There is still a need for further research and development in the field of mechanical CPR. In the future, it may be possible to improve the design and functionality of mechanical CPR devices, making them even more effective at restoring blood circulation and breathing in patients who have experienced cardiac arrest. Additionally, more research is needed to determine the optimal settings for mechanical CPR devices and to develop guidelines for their use in different patient populations. By continuing to advance the field of mechanical CPR, it may be possible to save even more lives in the future.

In conclusion, electromechanical CPR is a valuable tool for providing consistent and effective chest compressions in the event of cardiac arrest. It has the potential to improve survival rates and reduce the fatigue of rescuers.

Bibliography

[1] Mahmure Ayguin, Hacer Erten Yaman, ”Mechanical Chest Compression Devices: Historical Evolution, Classification and Current Practices, A Short Review”

[2] Md. Mujtabir Alam, Mahamud Hussain, Md. Ashik Amin ”Design of a Low-cost Automated Cardiopulmonary Resuscitation Device With Piston-Driven Chest Compression System” *4th International Conference on Electrical Engineering and Information and Communication Technology*

[3] Alejandro Mendoza Garcia, Stefan Eichhoorn, “Simulation of an Electro-Mechanical Resuscitation Device for Cardiopulmonary Resuscitation”

APPENDIX

- Program for Arduino
- Arduino Uno R3 - Datasheet
- MAX30100 - Datasheet

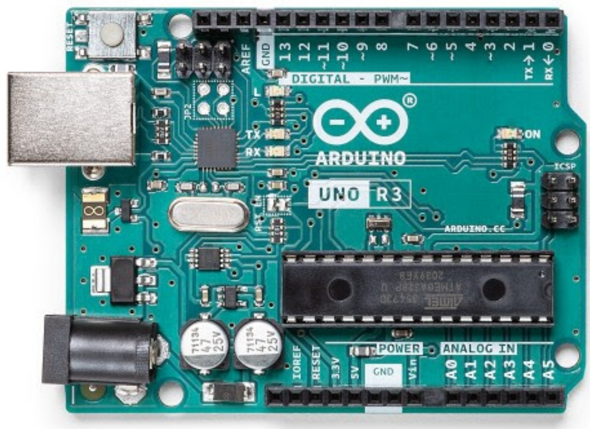
PROGRAM FOR ARDUINO

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#define REPORTING_PERIOD_MS 1000
LiquidCrystal_I2C lcd(0x27, 16, 2);
byte smile[] = {
B00000,
B00000,
B01010,
B00000,
B10001,
B01110,
B00000,
B00000
};
byte mod[] = {
B00000,
B00000,
B01010,
B00000,
B11111,
B00000,
B00000,
B00000
};
byte sad[] = {
B00000,
B00000,
```

```
B01010,
B00000,
B01110,
B10001,
B00000,
B00000
};
PulseOximeter pox;
uint32_t tsLastReport = 0;
void onBeatDetected()
{
Serial.println("Beat!!!");
}
void setup()
{
Serial.begin(115200);
lcd.init();
lcd.backlight();
lcd.createChar(1 , smile);
lcd.createChar(2 , mod);
lcd.createChar(3 , sad);
lcd.setCursor(0, 0);
lcd.print(" Pluse");
lcd.setCursor(0, 1);
lcd.print(" Oximeter");
delay(2000);
if (!pox.begin()) {
Serial.println("FAILED");
for (;;);
} else {
Serial.println("SUCCESS");
```

```
}
pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);
pox.setOnBeatDetectedCallback(onBeatDetected);
}
void loop()
{
pox.update();
if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
lcd.clear();
lcd.setCursor(0 , 0);
lcd.print("BPM : ");
lcd.print(pox.getHeartRate());
lcd.setCursor(0 , 1);
lcd.print("Sp02: ");
lcd.print(pox.getSpO2());
lcd.print("%");
tsLastReport = millis();
if (pox.getSpO2() >= 96) {
lcd.setCursor(15 , 1);
lcd.write(1);
}
else if (pox.getSpO2() <= 95 && pox.getSpO2() >= 91) {
lcd.setCursor(15 , 1);
lcd.write(2);
}
else if (pox.getSpO2() <= 90) {
lcd.setCursor(15 , 1);
lcd.write(3);
}
}
}
```

Product Reference Manual
SKU: A000066



Description

The Arduino UNO R3 is the perfect board to get familiar with electronics and coding. This versatile microcontroller is equipped with the well-known ATmega328P and the ATMega 16U2 Processor.
This board will give you a great first experience within the world of Arduino.

Target areas:

Maker, introduction, industries

Features

- **ATMega328P** Processor
  - **Memory**
    - AVR CPU at up to 16 MHz
    - 32KB Flash
    - 2KB SRAM
    - 1KB EEPROM
  - **Security**
    - Power On Reset (POR)
    - Brown Out Detection (BOD)
  - **Peripherals**
    - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
    - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
    - 1x USART with fractional baud rate generator and start-of-frame detection
    - 1x controller/peripheral Serial Peripheral Interface (SPI)
    - 1x Dual mode controller/peripheral I2C
    - 1x Analog Comparator (AC) with a scalable reference input
    - Watchdog Timer with separate on-chip oscillator
    - Six PWM channels
    - Interrupt and wake-up on pin change
  - **ATMega16U2 Processor**
    - 8-bit AVR® RISC-based microcontroller
  - **Memory**
    - 16 KB ISP Flash
    - 512B EEPROM
    - 512B SRAM
    - debugWIRE interface for on-chip debugging and programming
  - **Power**
    - 2.7-5.5 volts

CONTENTS

1 The Board **4**

1.1 Application Examples 4

1.2 Related Products 4

2 Ratings **4**

2.1 Recommended Operating Conditions 4

2.2 Power Consumption 5

3 Functional Overview **5**

3.1 Board Topology 5

3.2 Processor 6

3.3 Power Tree 6

4 Board Operation **7**

4.1 Getting Started - IDE 7

4.2 Getting Started - Arduino Web Editor 7

4.3 Getting Started - Arduino IoT Cloud 7

4.4 Sample Sketches 7

4.5 Online Resources 7

4.6 Board Recovery 8

5 Connector Pinouts **8**

5.1 JANALOG 9

5.2 JDIGITAL 9

5.3 Mechanical Information 10

5.4 Board Outline & Mounting Holes 10

6 Certifications **11**

6.1 Declaration of Conformity CE DoC (EU) 11

6.2 Declaration of Conformity to EU RoHS & REACH 211 01/19/2021 11

6.3 Conflict Minerals Declaration 12

7 FCC Caution **12**

8 Company Information **13**

9 Reference Documentation **13**

10 Revision History **13**

1 The Board

1.1 Application Examples

The UNO board is the flagship product of Arduino. Regardless if you are new to the world of electronics or will use the UNO as a tool for education purposes or industry-related tasks.

First entry to electronics: If this is your first project within coding and electronics, get started with our most used and documented board; Arduino UNO. It is equipped with the well-known ATmega328P processor, 14 digital input/output pins, 6 analog inputs, USB connections, ICSP header and reset button. This board includes everything you will need for a great first experience with Arduino.

Industry-standard development board: Using the Arduino UNO board in industries, there are a range of companies using the UNO board as the brain for their PLC's.

Education purposes: Although the UNO board has been with us for about ten years, it is still widely used for various education purposes and scientific projects. The board's high standard and top quality performance makes it a great resource to capture real time from sensors and to trigger complex laboratory equipment to mention a few examples.

1.2 Related Products

- Starter Kit
- Tinkerkit Braccio Robot
- Example

2 Ratings

2.1 Recommended Operating Conditions

Symbol	Description	Min	Max
	Conservative thermal limits for the whole board:	-40 °C (-40°F)	85 °C (185°F)

NOTE: In extreme temperatures, EEPROM, voltage regulator, and the crystal oscillator, might not work as expected due to the extreme temperature conditions

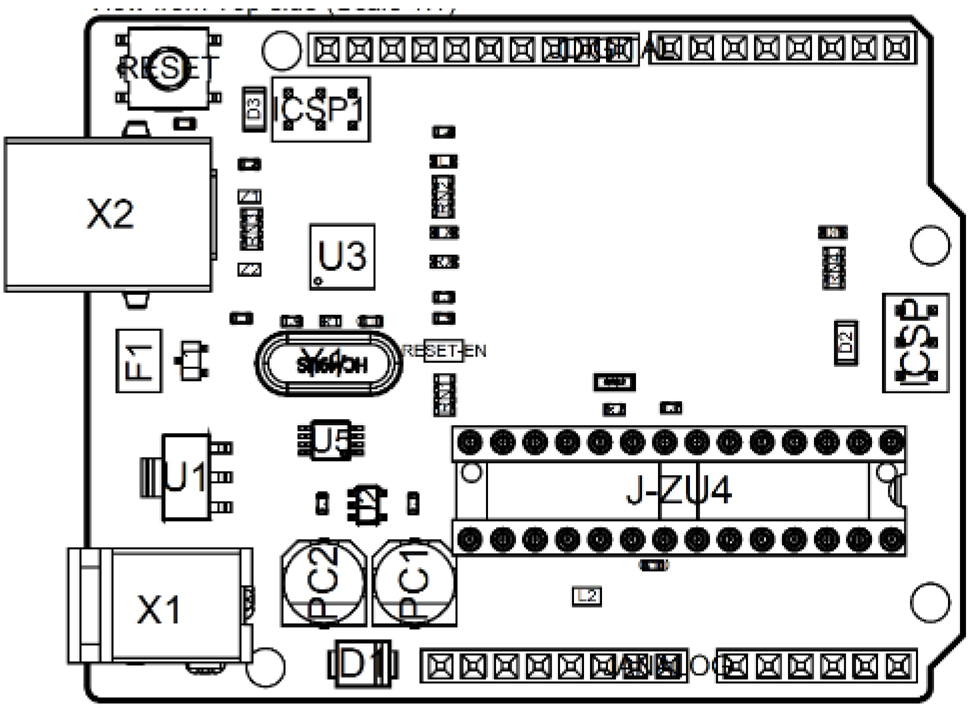
2.2 Power Consumption

Symbol	Description	Min	Typ	Max	Unit
VINMax	Maximum input voltage from VIN pad	6	-	20	V
VUSBMax	Maximum input voltage from USB connector		-	5.5	V
PMax	Maximum Power Consumption	-	-	xx	mA

3 Functional Overview

3.1 Board Topology

Top view



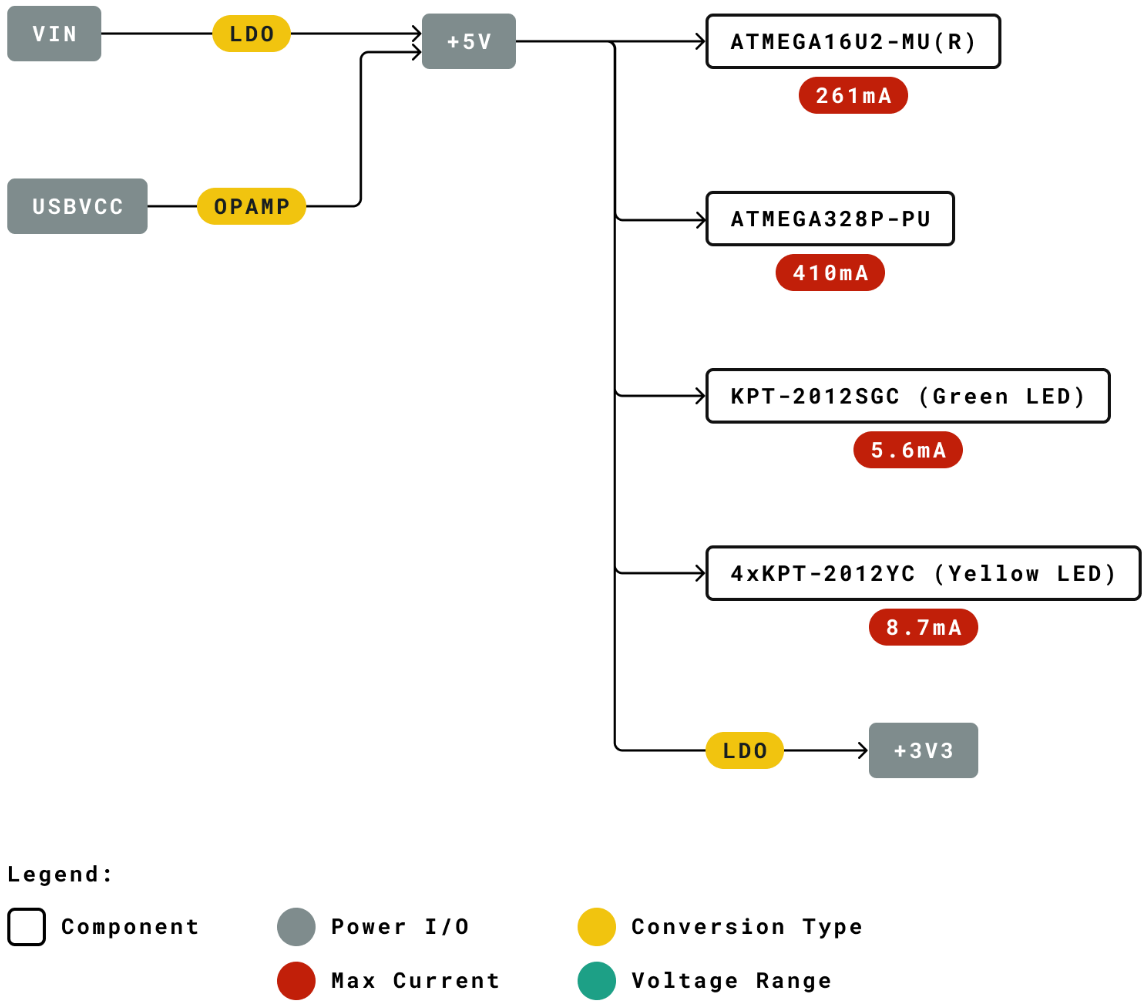
Board topology

Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through hole 6)
J-ZU4	ATMEGA328P Module	ICSP1	Pin header connector (through hole 6)
Y1	ECS-160-20-4X-DU Oscillator		

3.2 Processor

The Main Processor is a ATmega328P running at up tp 20 MHz. Most of its pins are connected to the external headers, however some are reserved for internal communication with the USB Bridge coprocessor.

3.3 Power Tree



Power tree

4 Board Operation

4.1 Getting Started - IDE

If you want to program your Arduino UNO while offline you need to install the Arduino Desktop IDE [1] To connect the Arduino UNO to your computer, you'll need a Micro-B USB cable. This also provides power to the board, as indicated by the LED.

4.2 Getting Started - Arduino Web Editor

All Arduino boards, including this one, work out-of-the-box on the Arduino Web Editor [2], by just installing a simple plugin.

The Arduino Web Editor is hosted online, therefore it will always be up-to-date with the latest features and support for all boards. Follow **[3]** to start coding on the browser and upload your sketches onto your board.

4.3 Getting Started - Arduino IoT Cloud

All Arduino IoT enabled products are supported on Arduino IoT Cloud which allows you to Log, graph and analyze sensor data, trigger events, and automate your home or business.

4.4 Sample Sketches

Sample sketches for the Arduino XXX can be found either in the “Examples” menu in the Arduino IDE or in the “Documentation” section of the Arduino Pro website [4]

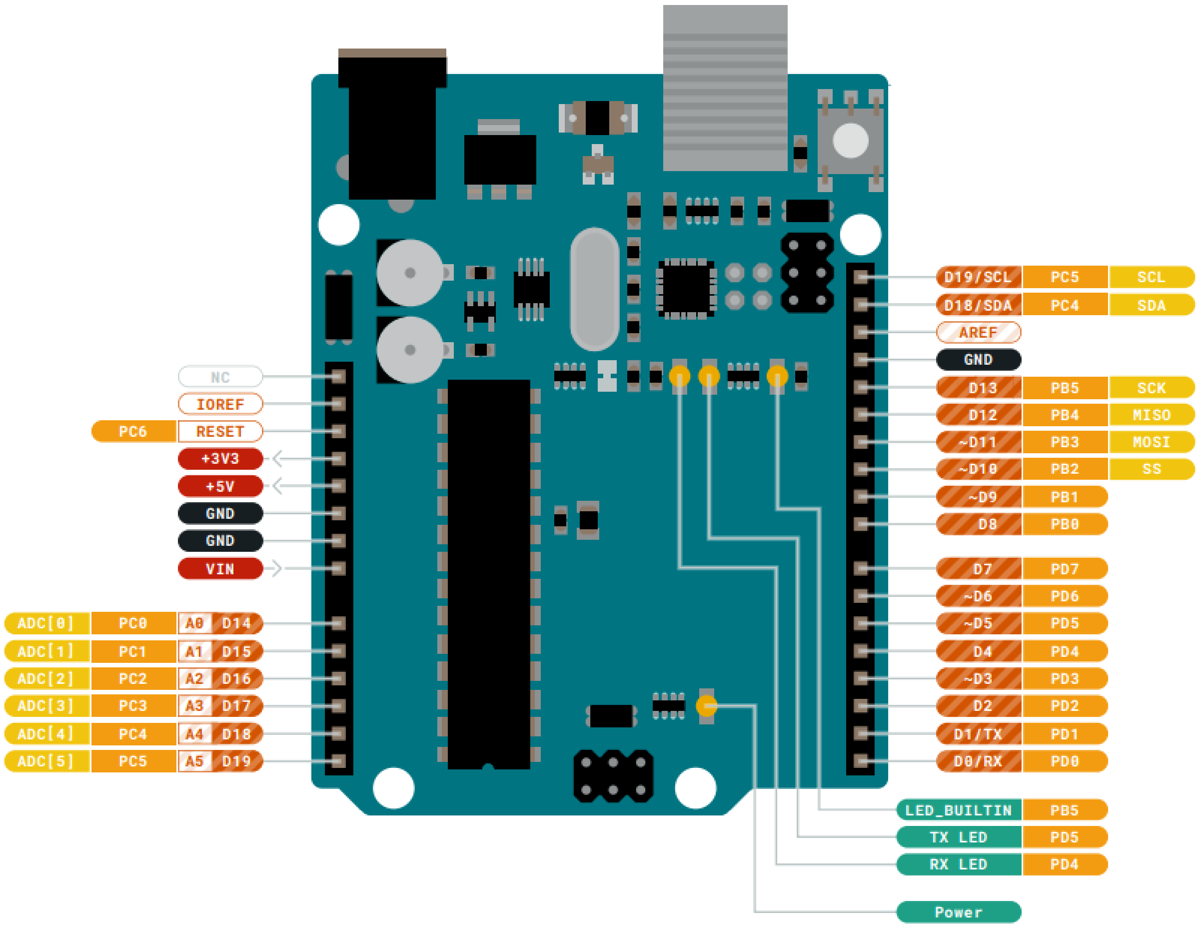
4.5 Online Resources

Now that you have gone through the basics of what you can do with the board you can explore the endless possibilities it provides by checking exciting projects on ProjectHub **[5]**, the Arduino Library Reference **[6]** and the online store **[7]** where you will be able to complement your board with sensors, actuators and more

4.6 Board Recovery

All Arduino boards have a built-in bootloader which allows flashing the board via USB. In case a sketch locks up the processor and the board is not reachable anymore via USB it is possible to enter bootloader mode by double-tapping the reset button right after power up.

5 Connector Pinouts



Pinout

5.1 JANALOG

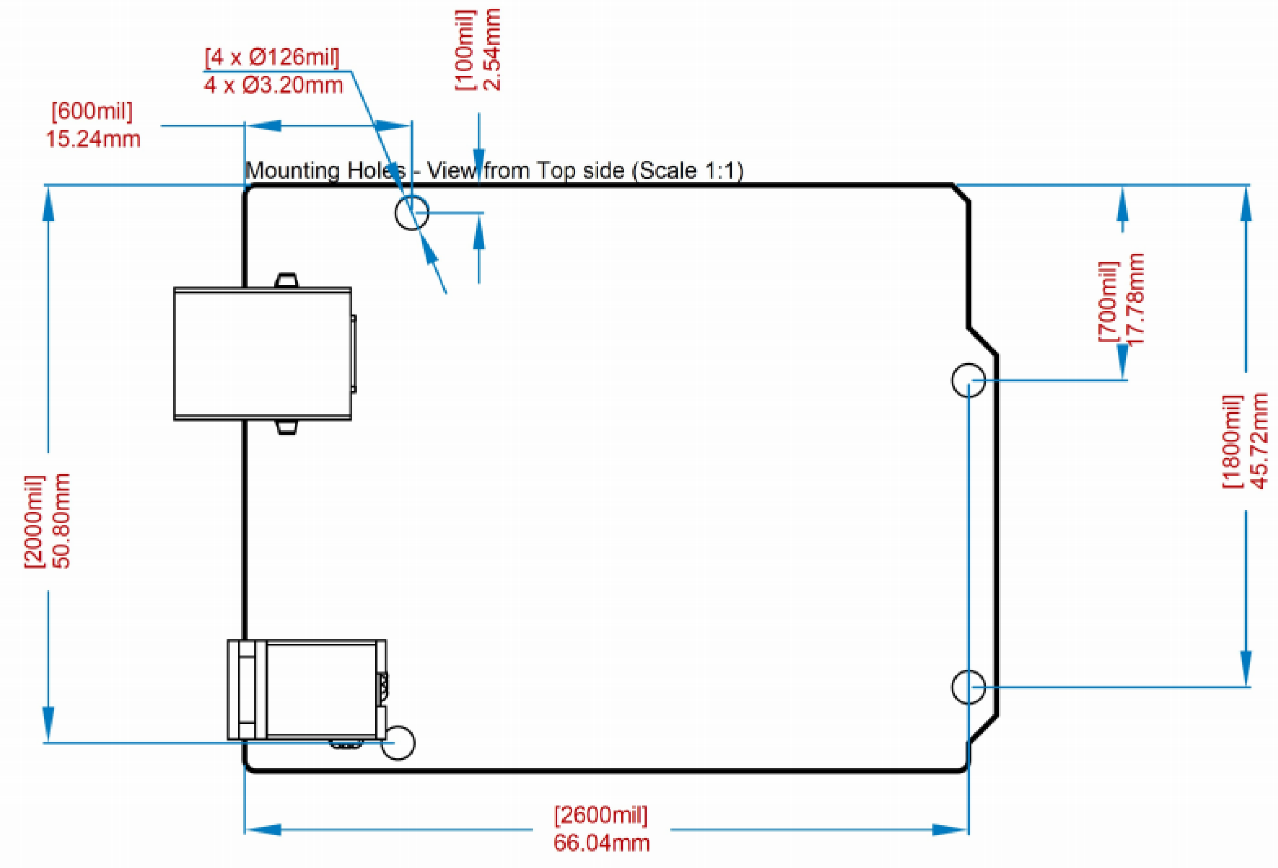
Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

5.2 JDIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

5.3 Mechanical Information

5.4 Board Outline & Mounting Holes



Board outline

6 Certifications

6.1 Declaration of Conformity CE DoC (EU)

We declare under our sole responsibility that the products above are in conformity with the essential requirements of the following EU Directives and therefore qualify for free movement within markets comprising the European Union (EU) and European Economic Area (EEA).

ROHS 2 Directive 2011/65/EU	
Conforms to:	EN50581:2012
Directive 2014/35/EU. (LVD)	
Conforms to:	EN 60950-1:2006/A11:2009/A1:2010/A12:2011/AC:2011
Directive 2004/40/EC & 2008/46/EC & 2013/35/EU, EMF	
Conforms to:	EN 62311:2008

6.2 Declaration of Conformity to EU RoHS & REACH 211 01/19/2021

Arduino boards are in compliance with RoHS 2 Directive 2011/65/EU of the European Parliament and RoHS 3 Directive 2015/863/EU of the Council of 4 June 2015 on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

Substance	Maximum limit (ppm)
Lead (Pb)	1000
Cadmium (Cd)	100
Mercury (Hg)	1000
Hexavalent Chromium (Cr6+)	1000
Poly Brominated Biphenyls (PBB)	1000
Poly Brominated Diphenyl ethers (PBDE)	1000
Bis(2-Ethylhexyl} phthalate (DEHP)	1000
Benzyl butyl phthalate (BBP)	1000
Dibutyl phthalate (DBP)	1000
Diisobutyl phthalate (DIBP)	1000

Exemptions: No exemptions are claimed.

Arduino Boards are fully compliant with the related requirements of European Union Regulation (EC) 1907 /2006 concerning the Registration, Evaluation, Authorization and Restriction of Chemicals (REACH). We declare none of the SVHCs (https://echa.europa.eu/web/guest/candidate-list-table), the Candidate List of Substances of Very High Concern for authorization currently released by ECHA, is present in all products (and also package) in quantities totaling in a concentration equal or above 0.1%. To the best of our knowledge, we also declare that our products do not contain any of the substances listed on the "Authorization List" (Annex XIV of the REACH regulations) and Substances of Very High Concern (SVHC) in any significant amounts as specified by the Annex XVII of Candidate list published by ECHA (European Chemical Agency) 1907 /2006/EC.

6.3 Conflict Minerals Declaration

As a global supplier of electronic and electrical components, Arduino is aware of our obligations with regards to laws and regulations regarding Conflict Minerals, specifically the Dodd-Frank Wall Street Reform and Consumer Protection Act, Section 1502. Arduino does not directly source or process conflict minerals such as Tin, Tantalum, Tungsten, or Gold. Conflict minerals are contained in our products in the form of solder, or as a component in metal alloys. As part of our reasonable due diligence Arduino has contacted component suppliers within our supply chain to verify their continued compliance with the regulations. Based on the information received thus far we declare that our products contain Conflict Minerals sourced from conflict-free areas.

7 FCC Caution

Any Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1) This device may not cause harmful interference

(2) this device must accept any interference received, including interference that may cause undesired operation.

FCC RF Radiation Exposure Statement:

1. This Transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.
2. This equipment complies with RF radiation exposure limits set forth for an uncontrolled environment.
3. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

English: User manuals for license-exempt radio apparatus shall contain the following or equivalent notice in a conspicuous location in the user manual or alternatively on the device or both. This device complies with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions:

(1) this device may not cause interference

(2) this device must accept any interference, including interference that may cause undesired operation of the device.

French: Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

(1) l' appareil nedoit pas produire de brouillage

(2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

IC SAR Warning:

English This equipment should be installed and operated with minimum distance 20 cm between the radiator and your body.

French: Lors de l' installation et de l' exploitation de ce dispositif, la distance entre le radiateur et le corps est d 'au moins 20 cm.

Important: The operating temperature of the EUT can't exceed 85℃ and shouldn't be lower than -40℃.

Hereby, Arduino S.r.l. declares that this product is in compliance with essential requirements and other relevant provisions of Directive 2014/53/EU. This product is allowed to be used in all EU member states.

8 Company Information

Company name	Arduino S.r.l
Company Address	Via Andrea Appiani 25 20900 MONZA Italy

9 Reference Documentation

Reference	Link
Arduino IDE (Desktop)	https://www.arduino.cc/en/Main/Software
Arduino IDE (Cloud)	https://create.arduino.cc/editor
Cloud IDE Getting Started	https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-4b3e4a
Arduino Pro Website	https://www.arduino.cc/pro
Project Hub	https://create.arduino.cc/projecthub?by=part&part_id=11332&sort=trending
Library Reference	https://www.arduino.cc/reference/en/
Online Store	https://store.arduino.cc/

10 Revision History

Date	Revision	Changes
xx/06/2021	1	Datasheet release

MAX30100 Pulse Oximeter and Heart-Rate Sensor IC for Wearable Health

General Description

The MAX30100 is an integrated pulse oximetry and heart-rate monitor sensor solution. It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals.

The MAX30100 operates from 1.8V and 3.3V power supplies and can be powered down through software with negligible standby current, permitting the power supply to remain connected at all times.

Applications

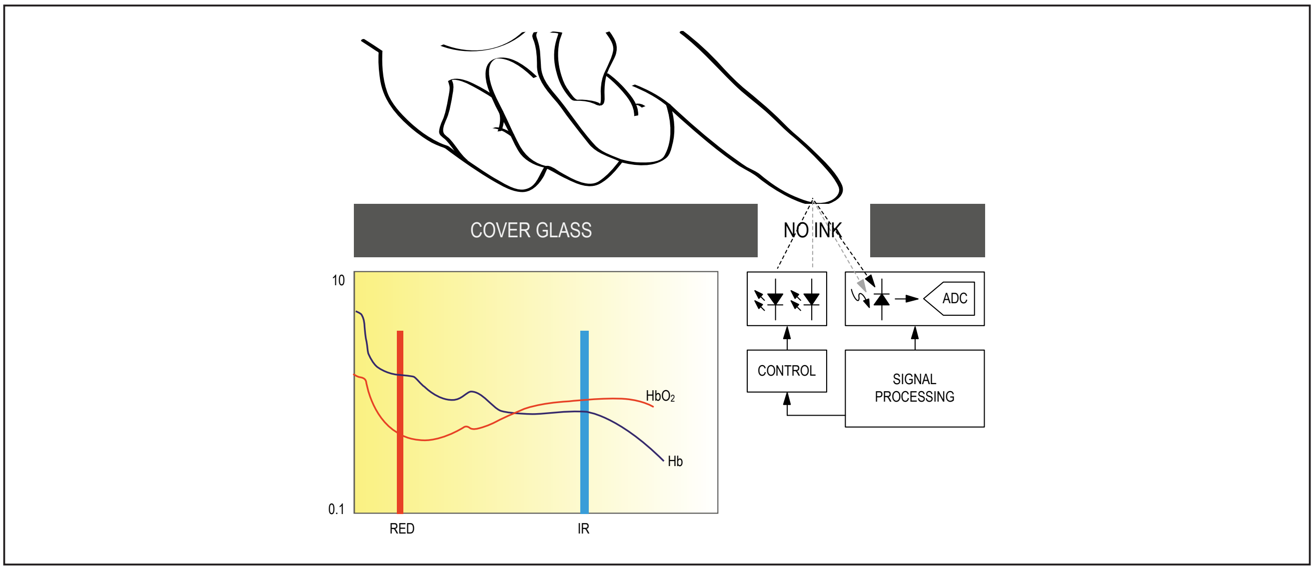
- Wearable Devices
- Fitness Assistant Devices
- Medical Monitoring Devices

Benefits and Features

- Complete Pulse Oximeter and Heart-Rate Sensor Solution Simplifies Design
  - Integrated LEDs, Photo Sensor, and High-Performance Analog Front -End
  - Tiny 5.6mm x 2.8mm x 1.2mm 14-Pin Optically Enhanced System-in-Package
- Ultra-Low-Power Operation Increases Battery Life for Wearable Devices
  - Programmable Sample Rate and LED Current for Power Savings
  - Ultra-Low Shutdown Current (0.7µA, typ)
- Advanced Functionality Improves Measurement Performance
  - High SNR Provides Robust Motion Artifact Resilience
  - Integrated Ambient Light Cancellation
  - High Sample Rate Capability
  - Fast Data Output Capability

Ordering Information ***appears at end of data sheet.***

System Block Diagram



Absolute Maximum Ratings

V_{DD} to GND ....-0.3V to +2.2V
GND to PGND ....-0.3V to +0.3V
x_DRV, x_LED+ to PGND ....-0.3V to +6.0V
All Other Pins to GND ....-0.3V to +6.0V
Output Short-Circuit Current Duration.......................Continuous
Continuous Input Current into Any Terminal ....±20mA
Continuous Power Dissipation (T_A = +70°C)
OESIP (derate 5.8mW/°C above +70°C) 464mW
Operating Temperature Range.......................... -40°C to +85°C
Soldering Temperature (reflow).....................................+260°C
Storage Temperature Range........................... -40°C to +105°C

Package Thermal Characteristics (Note 1)

OESIP
Junction-to-Ambient Thermal Resistance (θ_{JA}) ....150°C/W
Junction-to-Case Thermal Resistance (θ_{JC})............170°C/W

Note 1: Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to **www.maximintegrated.com/thermal-tutorial**.

Electrical Characteristics

(V_{DD} = 1.8V, V_{IR_LED+} = V_{R_LED+} = 3.3V, T_A = +25°C, min/max are from T_A = -40°C to +85°C, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
POWER SUPPLY						
Power-Supply Voltage	V_{DD}	Guaranteed by RED and IR count tolerance	1.7	1.8	2.0	V
LED Supply Voltage (R_LED+ or IR_LED+ to PGND)	V_{LED+}	Guaranteed by PSRR of LED Driver	3.1	3.3	5.0	V
Supply Current	I_{DD}	SpO_2 and heart rate modes, PW = 200µs, 50sps		600	1200	µA
		Heart rate only mode, PW = 200µs, 50sps		600	1200	
Supply Current in Shutdown	I_{SHDN}	T_A = +25°C, MODE = 0x80		0.7	10	µA
SENSOR CHARACTERISTICS						
ADC Resolution				14		bits
Red ADC Count (Note 3)	RED_C	Propriety ATE setup RED_PA = 0x05, LED_PW = 0x00, SPO2_SR = 0x07, T_A = +25°C	23,000	26,000	29,000	Counts
IR ADC Count (Note 3)	IR_C	Propriety ATE setup IR_PA = 0x09, LED_PW = 0x00, SPO2_SR = 0x07, T_A = +25°C	23,000	26,000	29,000	Counts
Dark Current Count	DC_C	RED_PA = IR_PA = 0x00, LED_PW = 0x03, SPO2_SR = 0x01		0	3	Counts
DC Ambient Light Rejection (Note 4)	ALR	Number of ADC counts with finger on sensor under direct sunlight (100K lux) LED_PW = 0x03, SPO2_SR = 0x01 — RED LED		0		Counts
		IR LED		0		

Electrical Characteristics (continued)

(V_{DD} = 1.8V, V_{IR_LED+} = V_{R_LED+} = 3.3V, T_A = +25°C, min/max are from T_A = -40°C to +85°C, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
IR ADC Count—PSRR (V_{DD})	$PSRR_{VDD}$	Propriety ATE setup 1.7V < V_{DD} < 2.0V, LED_PW = 0x03, SPO2_SR = 0x01, IR_PA = 0x09, IR_PA = 0x05, T_A = +25°C		0.25	2	%
		Frequency = DC to 100kHz, 100mV_{P-P}		10		LSB
RED/IR ADC Count—PSRR (X_LED+)	$PSRR_{LED}$	Propriety ATE setup 3.1V < X_LED+ < 5V, LED_PW = 0x03, SPO2_SR = 0x01, IR_PA = 0x09, IR_PA = 0x05, T_A = +25°C		0.05	2	%
		Frequency = DC to 100kHz, 100mV_{P-P}		10		LSB
ADC Integration Time	INT	LED_PW = 0x00		200		µs
		LED_PW = 0x03		1600		µs
IR LED CHARACTERISTICS (Note 4)						
LED Peak Wavelength	λ_P	I_{LED} = 20mA, T_A = +25°C	870	880	900	nm
Full Width at Half Max	$\Delta\lambda$	I_{LED} = 20mA, T_A = +25°C		30		nm
Forward Voltage	V_F	I_{LED} = 20mA, T_A = +25°C		1.4		V
Radiant Power	P_O	I_{LED} = 20mA, T_A = +25°C		6.5		mW
RED LED CHARACTERISTICS (Note 4)						
LED Peak Wavelength	λ_P	I_{LED} = 20mA, T_A = +25°C	650	660	670	nm
Full Width at Half Max	$\Delta\lambda$	I_{LED} = 20mA, T_A = +25°C		20		nm
Forward Voltage	V_F	I_{LED} = 20mA, T_A = +25°C		2.1		V
Radiant Power	P_O	I_{LED} = 20mA, T_A = +25°C		9.8		mW
TEMPERATURE SENSOR						
Temperature ADC Acquisition Time	T_T	T_A = +25°C		29		ms
Temperature Sensor Accuracy	T_A	T_A = +25°C		±1		°C
Temperature Sensor Minimum Range	T_{MIN}			-40		°C
Temperature Sensor Maximum Range	T_{MAX}			85		°C

Electrical Characteristics (continued)

(V_{DD} = 1.8V, V_{IR_LED+} = V_{R_LED+} = 3.3V, T_A = +25°C, min/max are from T_A = -40°C to +85°C, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
DIGITAL CHARACTERISTICS (SDA, SDA, $\overline{INT}$)						
Output Low Voltage SDA, $\overline{INT}$	V_{OL}	I_{SINK} = 6mA			0.4	V
I2C Input Voltage Low	V_{IL_I2C}	SDA, SCL			0.4	V
I2C Input Voltage High	V_{IH_I2C}	SDA, SCL	1.4			V
Input Hysteresis	V_{HYS}	SDA, SCL		200		mV
Input Capacitance	C_{IN}	SDA, SCL		10		pF
Input Leakage Current	I_{IN}	V_{IN} = 0V, T_A = +25°C (SDA, SCL, INT)		0.01	1	µA
		V_{IN} = 5.5V, T_A = +25°C (SDA, SCL, INT)		0.01	1	µA
I2C TIMING CHARACTERISTICS (SDA, SDA, $\overline{INT}$)						
I2C Write Address				AE		Hex
I2C Read Address				AF		Hex
Serial Clock Frequency	f_{SCL}		0		400	kHz
Bus Free Time Between STOP and START Conditions	t_{BUF}		1.3			µs
Hold Time (Repeated) START Condition	$t_{HD,START}$		0.6			µs
SCL Pulse-Width Low	t_{LOW}		1.3			µs
SCL Pulse-Width High	t_{HIGH}		0.6			µs
Setup Time for a Repeated START Condition	$t_{SU,START}$		0.6			µs
Data Hold Time	$t_{HD,DAT}$		0		900	ns
Data Setup Time	$t_{SU,DAT}$		100			ns
Setup Time for STOP Condition	$t_{SU,STOP}$		0.6			µs
Pulse Width of Suppressed Spike	t_{SP}		0		50	ns
Bus Capacitance	C_B				400	pF
SDA and SCL Receiving Rise Time	t_R		20 + 0.1C_B		300	ns
SDA and SCL Receiving Fall Time	t_{RF}		20 + 0.1C_B		300	ns
SDA Transmitting Fall Time	t_{TF}		20 + 0.1C_B		300	ns

Note 2: All devices are 100% production tested at T_A = +25°C. Specifications over temperature limits are guaranteed by Maxim Integrated's bench or proprietary automated test equipment (ATE) characterization.

Note 3: Specifications are guaranteed by Maxim Integrated's bench characterization and by 100% production test using proprietary ATE setup and conditions.

Note 4: For design guidance only. Not production tested.

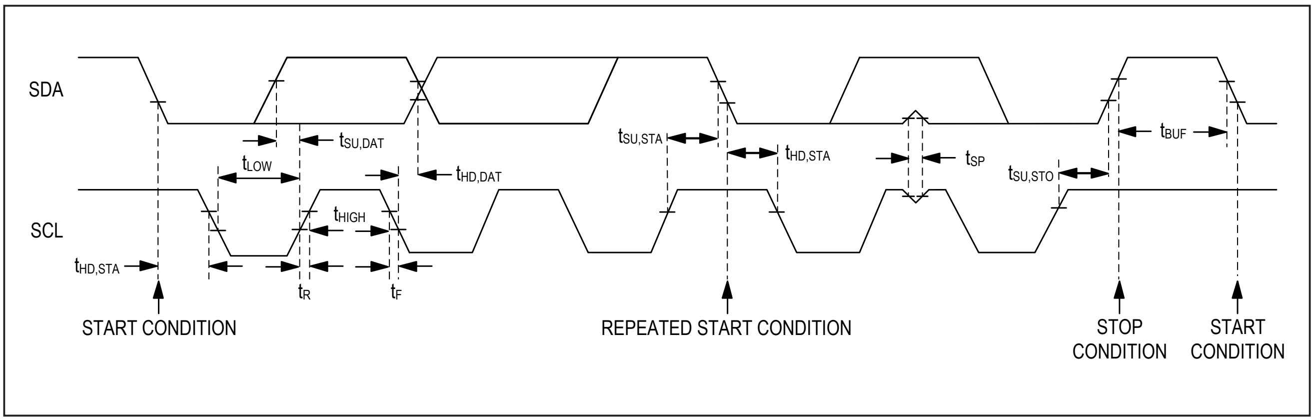
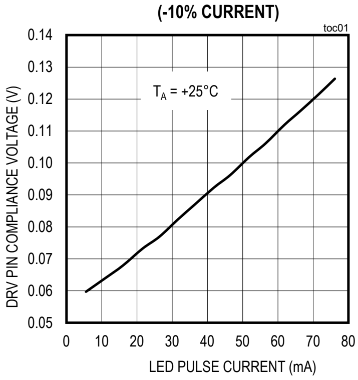


Figure 1. I²C-Compatible Interface Timing Diagram

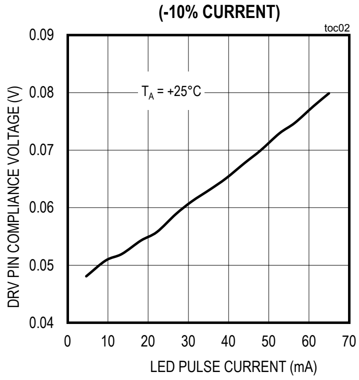
Typical Operating Characteristics

(V_{DD} = 1.8V, V_{IR_LED+} = V_{R_LED+} = 3.3V, T_A = +25°C, unless otherwise noted.)

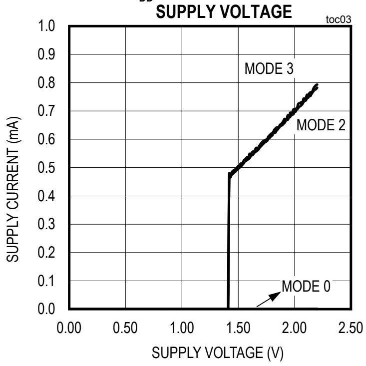
RED LED SUPPLY HEADROOM (-10% CURRENT)



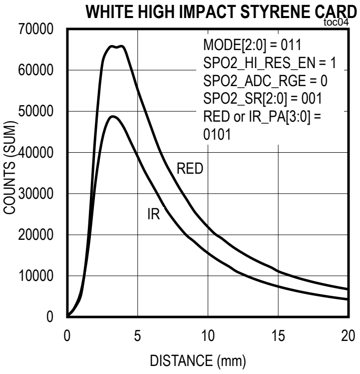
IR LED SUPPLY HEADROOM (-10% CURRENT)



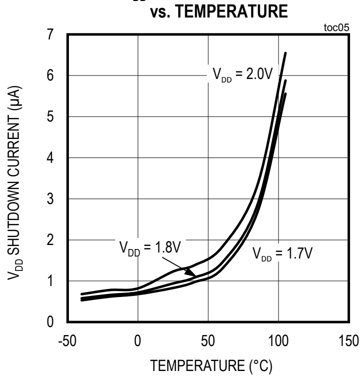
V_{DD} SUPPLY CURRENT vs. SUPPLY VOLTAGE



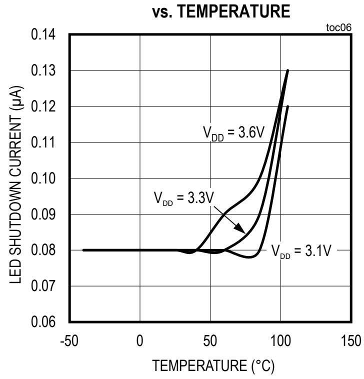
DC COUNTS vs. DISTANCE FOR WHITE HIGH IMPACT STYRENE CARD



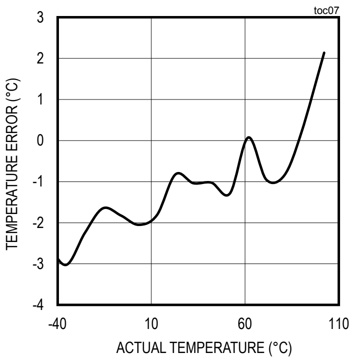
V_{DD} SHUTDOWN CURRENT vs. TEMPERATURE



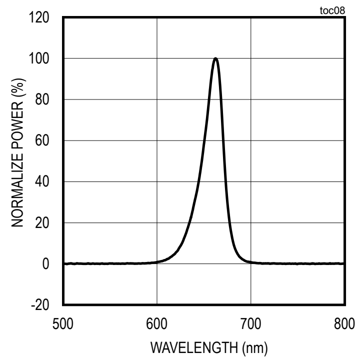
LED SHUTDOWN CURRENT vs. TEMPERATURE



ON-BOARD TEMPERATURE vs. ERROR

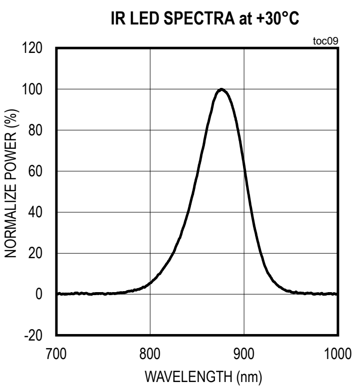


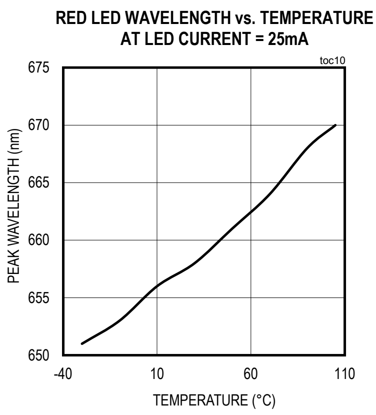
RED LED SPECTRA at +30°C

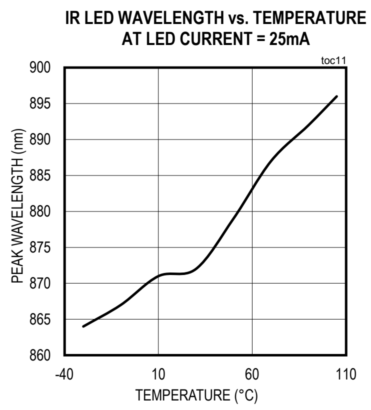


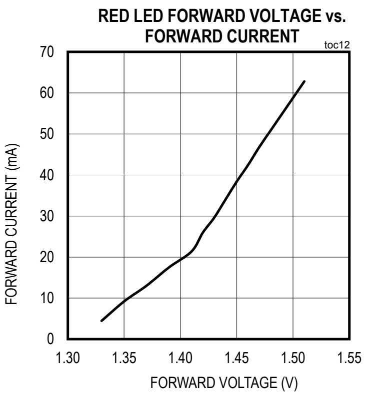
Typical Operating Characteristics (continued)

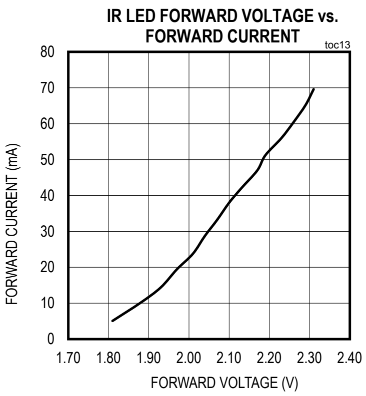
(V_{DD} = 1.8V, V_{IR_LED+} = V_{R_LED+} = 3.3V, T_A = +25°C, unless otherwise noted.)



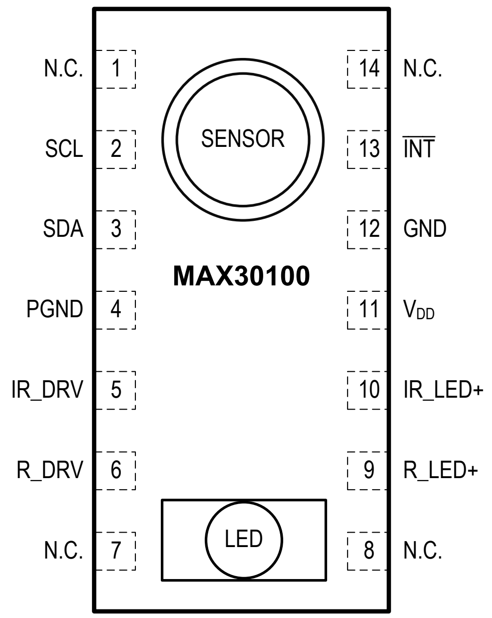








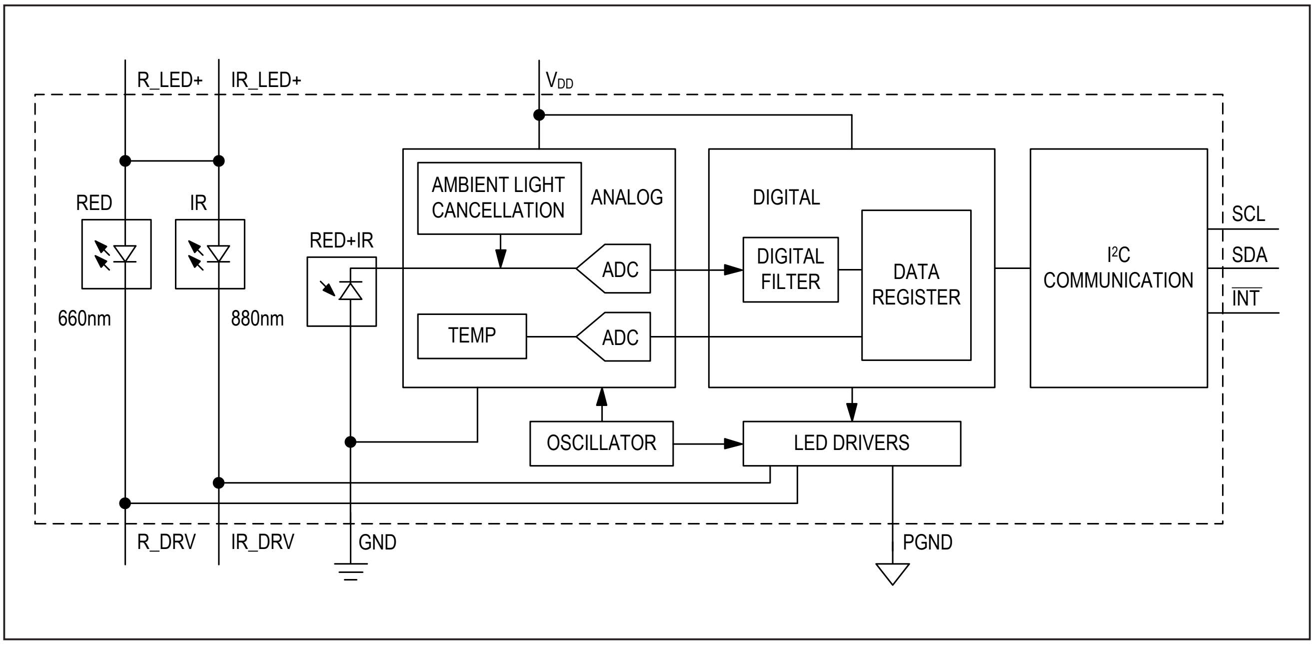
Pin Configuration



Pin Description

PIN	NAME	FUNCTION
1, 7, 8, 14	N.C.	No Connection. Connect to PCB Pad for Mechanical Stability.
2	SCL	I^2C Clock Input
3	SDA	I^2C Clock Data, Bidirectional (Open-Drain)
4	PGND	Power Ground of the LED Driver Blocks
5	IR_DRV	IR LED Cathode and LED Driver Connection Point. Leave floating in circuit.
6	R_DRV	Red LED Cathode and LED Driver Connection Point. Leave floating in circuit.
9	R_LED+	Power Supply (Anode Connection) for Red LED. Bypass to PGND for best performance. Connected to IR_LED+ internally.
10	IR_LED+	Power Supply (Anode Connection) for IR LED. Bypass to PGND for best performance. Connected to R_LED+ internally.
11	V_{DD}	Analog Power Supply Input. Bypass to GND for best performance.
12	GND	Analog Ground
13	$\overline{INT}$	Active-Low Interrupt (Open-Drain)

Functional Diagram



Detailed Description

The MAX30100 is a complete pulse oximetry and heart-rate sensor system solution designed for the demanding requirements of wearable devices. The MAX30100 provides very small total solution size without sacrificing optical or electrical performance. Minimal external hardware components are needed for integration into a wearable device.

The MAX30100 is fully configurable through software registers, and the digital output data is stored in a 16-deep FIFO within the device. The FIFO allows the MAX30100 to be connected to a microcontroller or microprocessor on a shared bus, where the data is not being read continuously from the device's registers.

SpO_2 Subsystem

The SpO_2 subsystem in the MAX30100 is composed of ambient light cancellation (ALC), 16-bit sigma delta ADC, and proprietary discrete time filter.

The SpO_2 ADC is a continuous time oversampling sigma delta converter with up to 16-bit resolution. The ADC output data rate can be programmed from 50Hz to 1kHz. The MAX30100 includes a proprietary discrete time filter to reject 50Hz/60Hz interference and low-frequency residual ambient noise.

Temperature Sensor

The MAX30100 has an on-chip temperature sensor for (optionally) calibrating the temperature dependence of the SpO_2 subsystem.

The SpO_2 algorithm is relatively insensitive to the wavelength of the IR LED, but the red LED's wavelength is critical to correct interpretation of the data. The temperature sensor data can be used to compensate the SpO_2 error with ambient temperature changes.

LED Driver

The MAX30100 integrates red and IR LED drivers to drive LED pulses for SpO_2 and HR measurements. The LED current can be programmed from 0mA to 50mA (typical only) with proper supply voltage. The LED pulse width can be programmed from 200µs to 1.6ms to optimize measurement accuracy and power consumption based on use cases.

Table 1. Register Maps and Descriptions

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
STATUS											
Interrupt Status	A_FULL	TEMP_ RDY	HR_RDY	SPO2_ RDY				PWR_ RDY	0x00	0X00	R
Interrupt Enable	ENB_A_ FULL	ENB_TE P_RDY	ENB_HR_ RDY	ENB_S O2_RDY					0x01	0X00	R/W
FIFO											
FIFO Write Pointer					FIFO_WR_PTR[3:0]				0x02	0x00	R/W
Over Flow Counter					OVF_COUNTER[3:0]				0x03	0x00	R/W
FIFO Read Pointer					FIFO_RD_PTR[3:0]				0x04	0x00	R/W
FIFO Data Register	FIFO_DATA[7:0]								0x05	0x00	R/W
CONFIGURATION											
Mode Configuration	SHDN	RESET			TEMP_ EN	MODE[2:0]			0x06	0x00	R/W
SPO2 Configuration		SPO2_HI_ RES_EN	RE- SERVED	SPO2_SR[2:0]			LED_PW[1:0]		0x07	0x00	R/W
RESERVED									0x08	0x00	R/W
LED Configuration	RED_PA[3:0]				IR_PA[3:0]				0x09	0x00	R/W
RESERVED									0x0A – 0x15	0x00	R/W
TEMPERATURE											
Temp_Integer	TINT[7:0]								0x16	0x00	R/W
Temp_Fraction					TFRAC[3:0]				0x17	0x00	R/W
RESERVED									0x8D	0x00	R/W
PART ID											
Revision ID	REV_ID[7:0]								0xFE	0xXX*	R
Part ID	PART_ID[7]								0xFF	0x11	R/W

*XX denotes any 2-digit hexidecimal number (00 to FF). Contact Maxim Integrated for the Revision ID number assigned for your product.

Interrupt Status (0x00)

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
Interrupt Status	A_FULL	TEMP_RDY	HR_RDY	SPO2_RDY				PWR_RDY	0x00	0X00	R

There are 5 interrupts and the functionality of each is exactly the same: pulling the active-low interrupt pin into its low state until the interrupt is cleared.

The interrupts are cleared whenever the interrupt status register is read, or when the register that triggered the interrupt is read. For example, if the SpO_2 sensor triggers an interrupt due to finishing a conversion, reading either the FIFO data register or the interrupt register clears the interrupt pin (which returns to its normal high state), and also clears all the bits in the interrupt status register to zero.

Bit 7: FIFO Almost Full Flag (A_FULL)

In SpO_2 and heart-rate modes, this interrupt triggers when the FIFO write pointer is the same as the FIFO read pointer minus one, which means that the FIFO has only one unwritten space left. If the FIFO is not read within the next conversion time, the FIFO becomes full and future data is lost.

Bit 6: Temperature Ready Flag (TEMP_RDY)

When an internal die temperature conversion is finished, this interrupt is triggered so the processor can read the temperature data registers.

Bit 5: Heart Rate Data Ready (HR_RDY)

In heart rate or SPO_2 mode, this interrupt triggers after every data sample is collected. A heart rate data sample consists of one IR data point only. This bit is automatically cleared when the FIFO data register is read.

Bit 4: SpO_2 Data Ready (SPO2_RDY)

In SpO_2 mode, this interrupt triggers after every data sample is collected. An SpO_2 data sample consists of one IR and one red data points. This bit is automatically cleared when the FIFO data register is read.

Bit 3: RESERVED

This bit should be ignored and always be zero in normal operation.

Bit 2: RESERVED

This bit should be ignored and always be zero in normal operation.

Bit 1: RESERVED

This bit should be ignored and always be zero in normal operation.

Bit 0: Power Ready Flag (PWR_RDY)

On power-up or after a brownout condition, when the supply voltage V_{DD} transitions from below the UVLO voltage to above the UVLO voltage, a power-ready interrupt is triggered to signal that the IC is powered up and ready to collect data.

Interrupt Enable (0x01)

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
Interrupt Enable	ENB_A_FULL	ENB_TEP_RDY	ENB_HR_RDY	ENB_SO2_RDY					0x01	0X00	R/W

Each source of hardware interrupt, with the exception of power ready, can be disabled in a software register within the MAX30100 IC. The power-ready interrupt cannot be disabled because the digital state of the MAX30100 is reset upon a brownout condition (low power-supply voltage), and the default state is that all the interrupts are disabled. It is important for the system to know that a brownout condition has occurred, and the data within the device is reset as a result.

When an interrupt enable bit is set to zero, the corresponding interrupt appears as 1 in the interrupt status register, but the $\overline{\text{INT}}$ pin is not pulled low.

The four unused bits (B3:B0) should always be set to zero (disabled) for normal operation.

FIFO (0x02–0x05)

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
FIFO Write Pointer					FIFO_WR_PTR[3:0]				0x02	0x00	R/W
Over Flow Counter					OVF_COUNTER[3:0]				0x03	0x00	R/W
FIFO Read Pointer					FIFO_RD_PTR[3:0]				0x04	0x00	R/W
FIFO Data Register	FIFO_DATA[7:0]								0x05	0x00	R/W

FIFO Write Pointer

The FIFO write pointer points to the location where the MAX30100 writes the next sample. This pointer advances for each sample pushed on to the FIFO. It can also be changed through the I^2C interface when MODE[2:0] is nonzero.

FIFO Overflow Counter

When the FIFO is full, samples are not pushed on to the FIFO, samples are lost. OVF_COUNTER counts the number of samples lost. It saturates at 0xF. When a complete sample is popped from the FIFO (when the read pointer advances), OVF_COUNTER is reset to zero.

FIFO Read Pointer

The FIFO read pointer points to the location from where the processor gets the next sample from the FIFO via the I^2C interface. This advances each time a sample is popped from the FIFO. The processor can also write to this pointer after reading the samples, which would allow rereading samples from the FIFO if there is a data communication error.

FIFO Data

The circular FIFO depth is 16 and can hold up to 16 samples of SpO_2 channel data (Red and IR). The FIFO_DATA register in the I^2C register map points to the next sample to be read from the FIFO. FIFO_RD_PTR points to this sample. Reading FIFO_DATA register does not automatically increment the register address; burst reading this register reads the same address over and over. Each sample is 4 bytes of data, so this register has to be read 4 times to get one sample.

The above registers can all be written and read, but in practice, only the FIFO_RD_PTR register should be written to in operation. The others are automatically incremented or filled with data by the MAX30100. When starting a new SpO_2

or heart-rate conversion, it is recommended to first clear the FIFO_WR_PTR, OVF_COUNTER, and FIFO_RD_PTR registers to all zeros (0x00) to ensure the FIFO is empty and in a known state. When reading the MAX30100 registers in one burst-read I^2C transaction, the register address pointer typically increments so that the next byte of data sent is from the next register, etc. The exception to this is the FIFO data register, register 0x05. When reading this register, the address pointer does not increment, but the FIFO_RD_PTR does. So the next byte of data sent will represent the next byte of data available in the FIFO.

Reading from the FIFO

Normally, reading registers from the I^2C interface autoincrements the register address pointer, so that all the registers can be read in a burst read without an I^2C restart event. In the MAX30100, this holds true for all registers except for the FIFO_DATA register (0x05).

Reading the FIFO_DATA register does not automatically increment the register address; burst reading this register reads the same address over and over. Each sample is 4 bytes of data, so this register has to be read 4 times to get one sample.

The other exception is 0xFF, reading more bytes after the 0xFF register does not advance the address pointer back to 0x00, and the data read is not meaningful.

FIFO Data Structure

The data FIFO consists of a 16-sample memory bank that stores both IR and RED ADC data. Since each sample consists of one IR word and one RED word, there are 4 bytes of data for each sample, and therefore, 64 total bytes of data can be stored in the FIFO. Figure 2 shows the structure of the FIFO graphically.

The FIFO data is left-justified as shown in Table 1; i.e. the MSB bit is always in the bit 15 position regardless of ADC resolution.

Each data sample consists of an IR and a red data word (2 registers), so to read one sample requires 4 I^2C byte reads in a row. The FIFO read pointer is automatically incremented after each 4-byte sample is read.

In heart-rate only mode, the 3rd and 4th bytes of each sample return zeros, but the basic structure of the FIFO remains the same.

Write/Read Pointers

Table 2. FIFO Data

ADC RESOLUTION	IR [15]	IR [14]	IR [13]	IR [12]	IR [11]	IR [10]	IR [9]	IR [8]	IR [7]	IR [6]	IR [5]	IR [4]	IR [3]	IR [2]	IR [1]	IR [0]
16-bit																
14-bit																
12-bit																
10-bit																

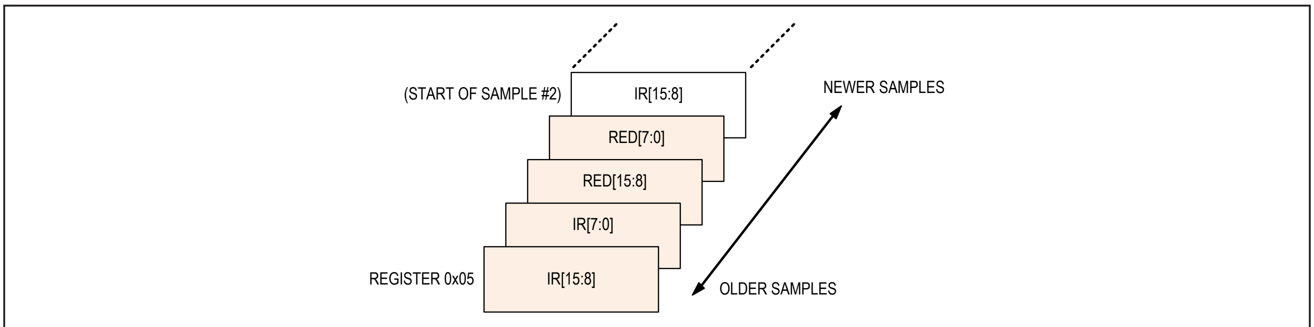


Figure 2. Graphical Representation of the FIFO Data Register

The locations to store new data, and the read pointer for reading data, are used to control the flow of data in the FIFO. The write pointer increments every time a new sample is added to the FIFO. The read pointer is incremented automatically every time a sample is read from the FIFO. To reread a sample from the FIFO, decrement its value by one and read the data register again.

The SpO_2 write/read pointers should be cleared (back to 0x0) upon entering SpO_2 mode or heart-rate mode, so that there is no old data represented in the FIFO. The pointers are not automatically cleared when changing modes, but they are cleared if V_{DD} is power cycled so that the V_{DD} voltage drops below its UVLO voltage.

Pseudo-Code Example of Reading Data from FIFO

First transaction: Get the FIFO_WR_PTR:

```
START;
Send device address + write mode
Send address of FIFO_WR_PTR;
REPEATED_START;
Send device address + read mode
Read FIFO_WR_PTR;
STOP;
```

The central processor evaluates the number of samples to be read from the FIFO:

```
NUM_AVAILABLE_SAMPLES = FIFO_WR_PTR – FIFO_RD_PTR
(Note: pointer wrap around should be taken into account)
NUM_SAMPLES_TO_READ = < less than or equal to NUM_AVAILABLE_SAMPLES >
```

Second transaction: Read `NUM_SAMPLES_TO_READ` **samples from the FIFO**:

```
START;
Send device address + write mode
Send address of FIFO_DATA;
REPEATED_START;
Send device address + read mode
for (i = 0; i < NUM_SAMPLES_TO_READ; i++) {
Read FIFO_DATA;
Save IR[15:8];
Read FIFO_DATA;
Save IR[7:0];
Read FIFO_DATA;
Save R[15:8];
Read FIFO_DATA;
Save R[7:0];
     }
STOP;
```

Third transaction: Write to FIFO_RD_PTR register. If the second transaction was successful, FIFO_RD_PTR points to the next sample in the FIFO, and this third transaction is not necessary. Otherwise, the processor updates the FIFO_RD_PTR appropriately, so that the samples are reread.

```
START;
Send device address + write mode
Send address of FIFO_RD_PTR;
Write FIFO_RD_PTR;
STOP;
```

Mode Configuration (0x06)

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
Mode Configuration	SHDN	RESET			TEMP_EN	MODE[2:0]			0x06	0x00	R/W

Bit 7: Shutdown Control (SHDN)

The part can be put into a power-save mode by setting this bit to one. While in power-save mode, all registers retain their values, and write/read operations function as normal. All interrupts are cleared to zero in this mode.

Bit 6: Reset Control (RESET)

When the RESET bit is set to one, all configuration, threshold, and data registers are reset to their power-on-state. The only exception is writing both RESET and TEMP_EN bits to one at the same time since temperature data registers 0x16 and 0x17 are not cleared. The RESET bit is cleared automatically back to zero after the reset sequence is completed.

Bit 3: Temperature Enable (TEMP_EN)

This is a self-clearing bit which, when set, initiates a single temperature reading from the temperature sensor. This bit is cleared automatically back to zero at the conclusion of the temperature reading when the bit is set to one in heart rate or SpO_2 mode.

Bits 2:0: Mode Control

These bits set the operating state of the MAX30100. Changing modes does not change any other setting, nor does it erase any previously stored data inside the data registers.

Table 3. Mode Control

MODE[2:0]	MODE
000	Unused
001	Reserved (Do not use)
010	HR only enabled
011	SPO_2 enabled
100–111	Unused

SpO2 Configuration (0x07)

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
SPO_2 Configuration		SPO2_HI_RES_EN	Reserved	SPO2_SR[2:0]			LED_PW[1:0]		0x07	0x00	R/W

Bit 6: SpO_2 High Resolution Enable (SPO2_HI_RES_EN)

Set this bit high. The SpO_2 ADC resolution is 16-bit with 1.6ms LED pulse width.

Bit 5: Reserved. Set low (default).

Bit 4:2: SpO_2 Sample Rate Control

These bits define the effective sampling rate, with one sample consisting of one IR pulse/conversion and one RED pulse/conversion.

The sample rate and pulse width are related, in that the sample rate sets an upper bound on the pulse width time. If the user selects a sample rate that is too high for the selected LED_PW setting, the highest possible sample rate will instead be programmed into the register.

Bits 1:0: LED Pulse Width Control

These bits set the LED pulse width (the IR and RED have the same pulse width), and therefore, indirectly set the integration time of the ADC in each sample. The ADC resolution is directly related to the integration time.

Table 4. SpO_2 Sample Rate Control

SPO2_SR[2:0]	SAMPLES (PER SECOND)
000	50
001	100
010	167
011	200
100	400
101	600
110	800
111	1000

Table 5. LED Pulse Width Control

LED_PW[1:0]	PULSE WIDTH (µs)	ADC RESOLUTION (BITS)
00	200	13
01	400	14
10	800	15
11	1600	16

LED Configuration (0x09)

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
LED Configuration	RED_PA[3:0]				IR_PA[3:0]				0x09	0x00	R/W

Bits 7:4: Red LED Current Control

These bits set the current level of the Red LED as in Table 6.

Bits 3:0: IR LED Current Control

These bits set the current level of the IR LED as in Table 6.

Table 6. LED Current Control

Red_PA[3:0] OR IR_PA[3:0]	TYPICAL LED CURRENT (mA)*
0000	0.0
0001	4.4
0010	7.6
0011	11.0
0100	14.2
0101	17.4
0110	20.8
0111	24.0
1000	27.1
1001	30.6
1010	33.8
1011	37.0
1100	40.2
1101	43.6
1110	46.8
1111	50.0

*Actual measured LED current for each part can vary widely due to the proprietary trim methodology.

Temperature Data (0x16–0x17)

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
Temp_Integer	TINT[7:0]								0x16	0x00	R/W
Temp_Fraction					TFRAC[3:0]				0x17	0x00	R/W

Temperature Integer

The on-board temperature ADC output is split into two registers, one to store the integer temperature and one to store the fraction. Both should be read when reading the temperature data, and the following equation shows how to add the two registers together:

$$T_{MEASURED} = T_{INTEGER} + T_{FRACTION}$$

This register stores the integer temperature data in two's complement format, where each bit corresponds to degree Celsius.

Table 7. Temperature Integer

REGISTER VALUE (hex)	TEMPERATURE (°C)
0x00	0
0x00	+1
...	...
0x7E	+126
0x7F	+127
0x80	-128
0x81	-127
...	...
0xFE	-2
0xFF	-1

Temperature Fraction

This register stores the fractional temperature data in increments of 0.0625°C (1/16th of a degree).

If this fractional temperature is paired with a negative integer, it still adds as a positive fractional value (e.g., -128°C + 0.5°C = -127.5°C).

Applications Information

Sampling Rate and Performance

The MAX30100 ADC is a 16-bit sigma delta converter. The ADC sampling rate can be configured from 50sps to 1ksps. The maximum sample rate for the ADC depends on the selected pulse width, which in turn, determines the ADC resolution. For instance, if the pulse width is set to 200μs, then the ADC resolution is 13 bits and all sample rates from 50sps to 1ksps are selectable. However, if the pulse width is set to 1600μs, then only sample rates of 100sps and 50sps can be set. The allowed sample rates for both SpO_2 and HR mode are summarized in Table 8 and Table 9.

Power Considerations

The LEDs in MAX30100 are pulsed with a low duty cycle for power savings, and the pulsed currents can cause ripples in the LED power supply. To ensure these pulses do not translate into optical noise at the LED outputs, the power supply must be designed to handle peak LED current. Ensure that the resistance and inductance from the power supply (battery, DC/DC converter, or LDO) to the device LED+ pins is much smaller than 1Ω, and that there is at least 1μF of power-supply bypass capacitance to a low impedance ground plane. The decoupling capacitor should be located physically as close as possible to the MAX30100 device.

In the heart-rate only mode, the red LED is inactive, and only the IR LED is used to capture optical data and determine the heart rate. This mode allows power savings due to the red LED being off; in addition, the IR_LED+ power supply can be reduced to save power because the forward voltage of the IR LED is significantly less than that of the red LED.

The average I_{DD} and LED current as function of pulse width and sampling rate is summarized in Table 10 to Table 13.

Table 8. SpO_2 Mode (Allowed Settings)

SAMPLES (per second)	PULSE WIDTH (μs)			
	200	400	800	1600
50	O	O	O	O
100	O	O	O	O
167	O	O	O	
200	O	O	O	
400	O	O		
600	O			
800	O			
1000	O			
Resolution (bits)	13	14	15	16

Table 9. Heart-Rate Mode (Allowed Settings)

SAMPLES (per second)	PULSE WIDTH (μs)			
	200	400	800	1600
50	O	O	O	O
100	O	O	O	O
167	O	O	O	
200	O	O	O	
400	O	O		
600	O	O		
800	O	O		
1000	O	O		
Resolution (bits)	13	14	15	16

Table 10. SpO_2 Mode: Average IDD Current (µA) R_PA = 0x3, IR_PA = 0x3

SAMPLES (per second)	PULSE WIDTH (µs)			
	200	400	800	1600
50	628	650	695	782
100	649	691	776	942
167	678	748	887	
200	692	775	940	
400	779	944		
600	865			
800	952			
1000	1037			

Table 11. SpO_2 Mode: Average LED Current (mA) R_PA = 0x3, IR_PA = 0x3

SAMPLES (per second)	PULSE WIDTH (µs)			
	200	400	800	1600
50	0.667	1.332	2.627	5.172
100	1.26	2.516	4.96	9.766
167	2.076	4.145	8.173	
200	2.491	4.93	9.687	
400	4.898	9.765		
600	7.319			
800	9.756			
1000	12.17			

Table 12. Heart-Rate Mode: Average IDD Current (µA) IR_PA = 0x3

SAMPLES (per second)	PULSE WIDTH (µs)			
	200	400	800	1600
50	608	616	633	667
100	617	634	669	740
167	628	658	716	831
200	635	670	739	876
400	671	740	878	
600	707	810		
800	743	881		
1000	779	951		

Table 13. Heart-Rate Mode: Average LED Current (mA) IR_PA = 0x3

SAMPLES (per second)	PULSE WIDTH (µs)			
	200	400	800	1600
50	0.256	0.511	1.020	2.040
100	0.512	1.022	2.040	4.077
167	0.854	1.705	3.404	6.795
200	1.023	2.041	4.074	8.130
400	2.042	4.074	8.123	
600	3.054	6.089		
800	4.070	8.109		
1000	5.079	10.11		

Hardware Interrupt

The active-low interrupt pin pulls low when an interrupt is triggered. The pin is open-drain and requires a pullup resistor or current source to an external voltage supply (up to +5V from GND). The interrupt pin is not designed to sink large currents, so the pullup resistor value should be large, such as 4.7kΩ.

The internal FIFO stores up to 16 samples, so that the system processor does not need to read the data after every sample. Temperature data may be needed to properly interpret SpO_2 data, but the temperature does not need to be sampled very often—once a second or every few seconds should be sufficient. In heart-rate mode temperature information is not necessary.

Table 14. Red LED Current Settings vs. LED Temperature Rise

RED LED CURRENT SETTING	RED LED DUTY CYCLE (% OF LED PULSE WIDTH TO SAMPLE TIME)	ESTIMATED TEMPERATURE RISE (ADD TO TEMPERATURE SENSOR MEASUREMENT) (°C)
0001 (3.1mA)	8	0.1
1111 (35mA)	8	2
0001 (3.1mA)	16	0.3
1111 (35mA)	16	4
0001 (3.1mA)	32	0.6
1111 (35mA)	32	8

Timing for Measurements and Data Collection

Timing in SpO_2 Mode

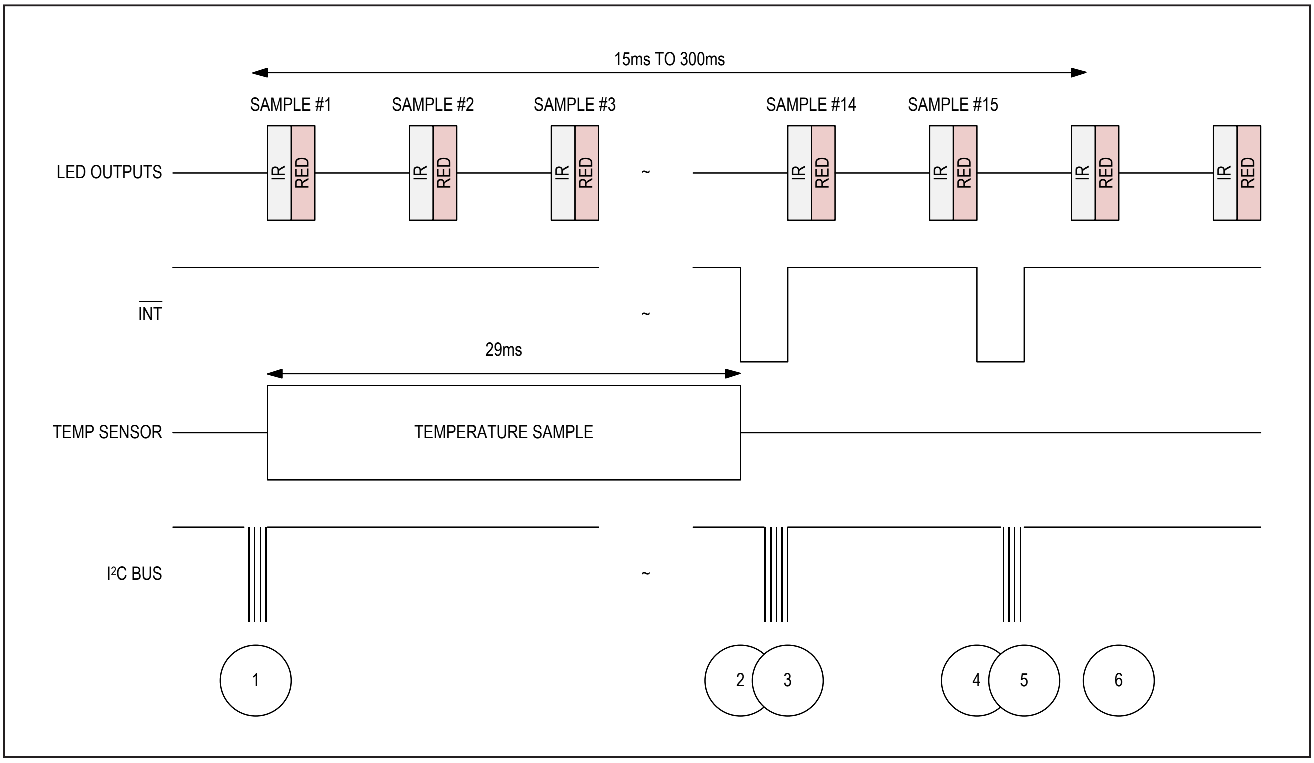


Figure 3. Timing for Data Acquisition and Communication When in SpO_2 Mode

Table 15. Events Sequence for Figure 3 in SpO_2 Mode

EVENT	DESCRIPTION	COMMENTS
1	Enter into SpO_2 mode. Initiate a temperature measurement.	I^2C Write Command Sets MODE[2:0] = 0x03. At the same time, set the TEMP_EN bit to initiate a single temperature measurement. Mask the SPO2_RDY Interrupt.
2	Temperature measurement complete, interrupt generated	TEMP_RDY interrupt triggers, alerting the central processor to read the data.
3	Temp data is read, interrupt cleared	
4	FIFO is almost full, interrupt generated	Interrupt is generated when the FIFO has only one empty space left.
5	FIFO data is read, interrupt cleared	
6	Next sample is stored	New sample is stored at the new read pointer location. Effectively, it is now the first sample in the FIFO.

Timing in Heart-Rate Mode

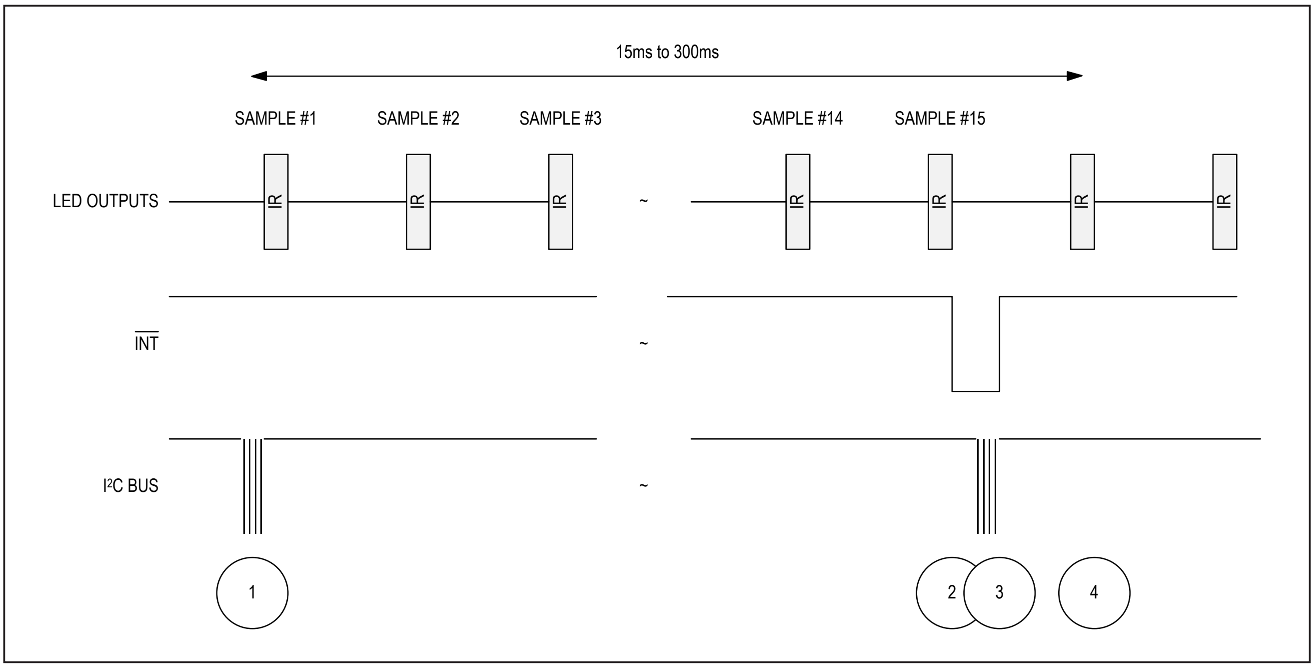


Figure 4. Timing for Data Acquisition and Communication When in Heart Rate Mode

Table 16. Events Sequence for Figure 4 in Heart-Rate Mode

EVENT	DESCRIPTION	COMMENTS
1	Enter into heart rate mode	I^2C Write Command Sets MODE[2:0] = 0x02. Mask the HR_RDY interrupt.
2	FIFO is almost full, interrupt generated	Interrupt is generated when the FIFO has only one empty space left.
3	FIFO data is read, interrupt cleared	
4	Next sample is stored	New sample is stored at the new read pointer location. Effectively, it is now the first sample in the FIFO.

Power Sequencing and Requirements

Power-Up Sequencing

Figure 5 shows the recommended power-up sequence for the MAX30100.

It is recommended to power the V_{DD} supply first, before the LED power supplies (R_LED+, IR_LED+). The interrupt and I^2C pins can be pulled up to an external voltage even when the power supplies are not powered up.

After the power is established, an interrupt occurs to alert the system that the MAX30100 is ready for operation. Reading the I^2C interrupt register clears the interrupt, as shown in Figure 5.

Power-Down Sequencing

The MAX30100 is designed to be tolerant of any power-supply sequencing on power-down.

I^2C Interface

The MAX30100 features an I^2C/SMBus-compatible, 2-wire serial interface consisting of a serial data line (SDA) and a serial clock line (SCL). SDA and SCL facilitate communication between the MAX30100 and the master at clock rates up to 400kHz. Figure 1 shows the 2-wire interface timing diagram. The master generates SCL and initiates data transfer on the bus. The master device writes data to the MAX30100 by transmitting the proper slave address followed by data. Each transmit sequence is framed by a START (S) or REPEATED START (Sr) condition and a STOP (P) condition. Each word transmitted to the MAX30100 is 8 bits long and is followed by an acknowledge clock pulse. A master reading data from the MAX30100 transmits the proper slave address followed by a series of nine SCL pulses.

The MAX30100 transmits data on SDA in sync with the master-generated SCL pulses. The master acknowledges receipt of each byte of data. Each read sequence is framed by a START (S) or REPEATED START (Sr) condition, a not acknowledge, and a STOP (P) condition. SDA operates as both an input and an open-drain output. A pullup resistor, typically greater than 500Ω, is required on SDA. SCL operates only as an input. A pullup resistor, typically greater than 500Ω, is required on SCL if there are multiple masters on the bus, or if the single master has an open-drain SCL output.

Bit Transfer

One data bit is transferred during each SCL cycle. The data on SDA must remain stable during the high period of the SCL pulse. Changes in SDA while SCL is high are control signals. See the *START and STOP Conditions* section.

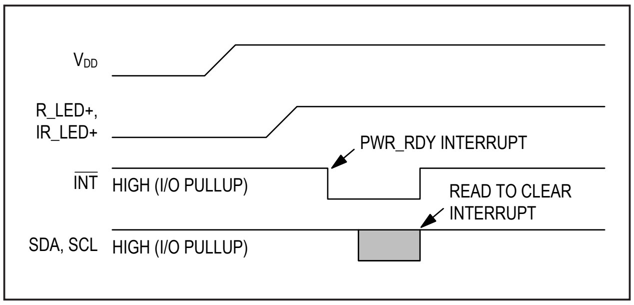


Figure 5. Power-Up Sequence of the Power-Supply Rails

START and STOP Conditions

SDA and SCL idle high when the bus is not in use. A master initiates communication by issuing a START condition. A START condition is a high-to-low transition on SDA with SCL high. A STOP condition is a low-to-high transition on SDA while SCL is high (Figure 6). A START condition from the master signals the beginning of a transmission to the MAX30100. The master terminates transmission, and frees the bus, by issuing a STOP condition. The bus remains active if a REPEATED START condition is generated instead of a STOP condition.

Early STOP Conditions

The MAX30100 recognizes a STOP condition at any point during data transmission except if the STOP condition occurs in the same high pulse as a START condition. For proper operation, do not send a STOP condition during the same SCL high pulse as the START condition.

Slave Address

A bus master initiates communication with a slave device by issuing a START condition followed by the 7-bit slave ID. When idle, the MAX30100 waits for a START condition followed by its slave ID. The serial interface compares each slave ID bit by bit, allowing the interface to power down and disconnect from SCL immediately if an incorrect slave ID is detected. After recognizing a START condition followed by the correct slave ID, the MAX30100 is ready to accept or send data. The LSB of the slave ID word is the Read/$\overline{\text{Write}}$ ($R/\overline{W}$) bit. $R/\overline{W}$ indicates whether the master is writing to or reading data from the MAX30100. $R/\overline{W}$ = 0 selects a write condition, $R/\overline{W}$ = 1 selects a read condition). After receiving the proper slave ID, the MAX30100 issues an ACK by pulling SDA low for one clock cycle.

The MAX30100 slave ID consists of seven fixed bits, B7–B1 (set to 0b1010111). The most significant slave ID bit (B7) is transmitted first, followed by the remaining bits. Table 18 shows the possible slave IDs of the device.

Acknowledge

The acknowledge bit (ACK) is a clocked 9th bit that the MAX30100 uses to handshake receipt each byte of data when in write mode (Figure 7). The MAX30100 pulls down SDA during the entire master-generated 9th clock pulse if the previous byte is successfully received. Monitoring ACK allows for detection of unsuccessful data transfers. An unsuccessful data transfer occurs if a receiving device is busy or if a system fault has occurred. In the event of an unsuccessful data transfer, the bus master will retry communication. The master pulls down SDA during the 9th clock cycle to acknowledge receipt of data when the MAX30100 is in read mode. An acknowledge is sent by the master after each read byte to allow data transfer to continue. A not-acknowledge is sent when the master reads the final byte of data from the MAX30100, followed by a STOP condition.

Table 17. Slave ID Description

B7	B6	B5	B4	B3	B2	B1	B0	WRITE ADDRESS	READ ADDRESS
1	0	1	0	1	1	1	$R/\overline{W}$	0xAE	0xAF

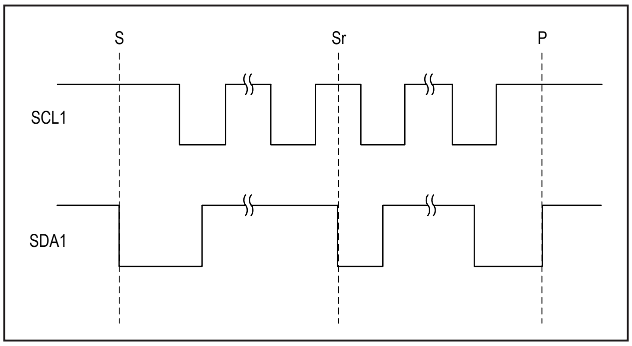


Figure 6. START, STOP, and REPEATED START Conditions

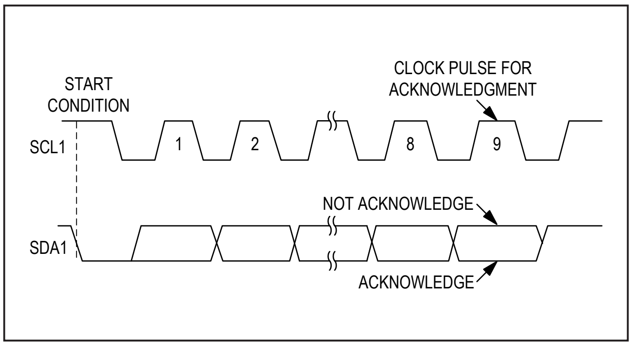


Figure 7. Acknowledge

Write Data Format

For the write operation, send the slave ID as the first byte followed by the register address byte and then one or more data bytes. The register address pointer increments automatically after each byte of data received. For example, the entire register bank can be written by at one time. Terminate the data transfer with a STOP condition. The write operation is shown in Figure 8.

The internal register address pointer increments automatically, so writing additional data bytes fill the data registers in order.

Read Data Format

For the read operation, two I2C operations must be performed. First, the slave ID byte is sent followed by the I2C register that you wish to read. Then a REPEATED START (Sr) condition is sent, followed by the read slave ID. The MAX30100 then begins sending data beginning with the register selected in the first operation. The read pointer increments automatically, so the MAX30100 continues sending data from additional registers in sequential order until a STOP (P) condition is received. The exception to this is the FIFO_DATA register, at which the read pointer no longer increments when reading additional bytes. To read the next register after FIFO_DATA, an I2C write command is necessary to change the location of the read pointer.

An initial write operation is required to send the read register address.

Data is sent from registers in sequential order, starting from the register selected in the initial I2C write operation. If the FIFO_DATA register is read, the read pointer does not automatically increment, and subsequent bytes of data contain the contents of the FIFO.

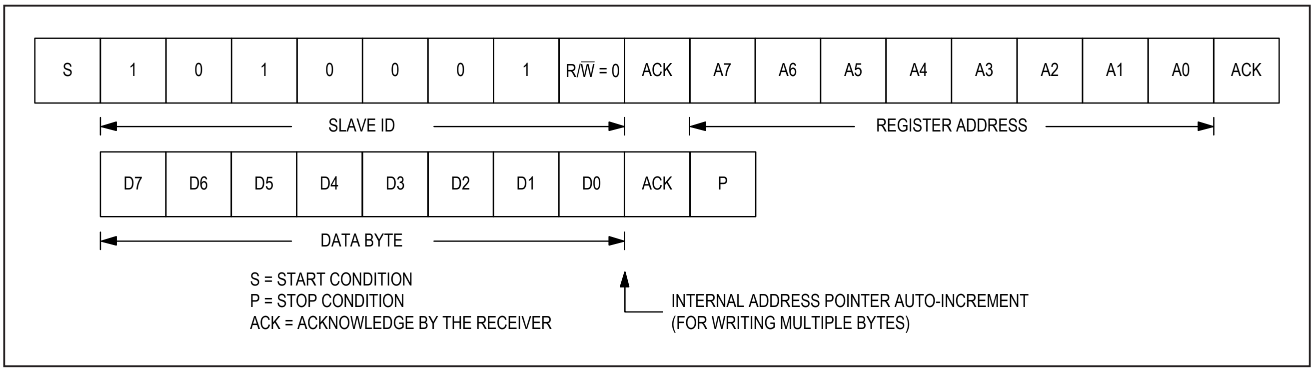


Figure 8. Writing One Data Byte to the MAX30100

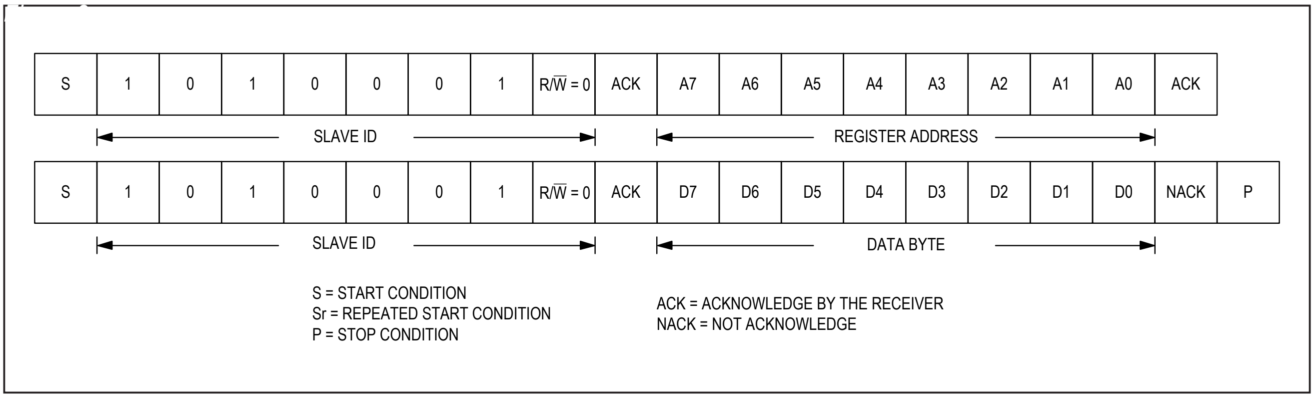


Figure 9. Reading One Byte of Data from the MAX30100

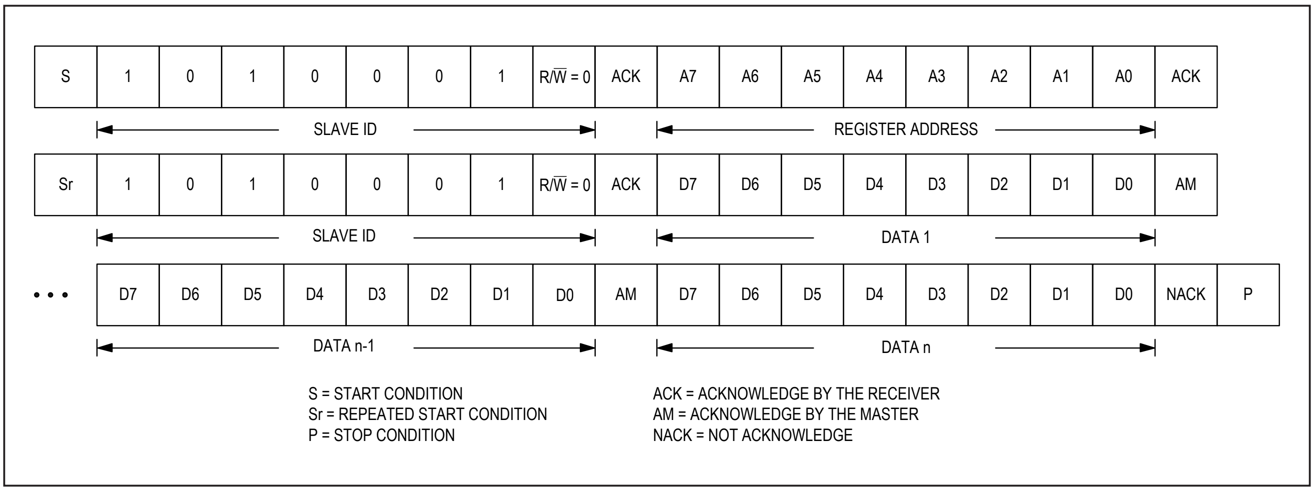
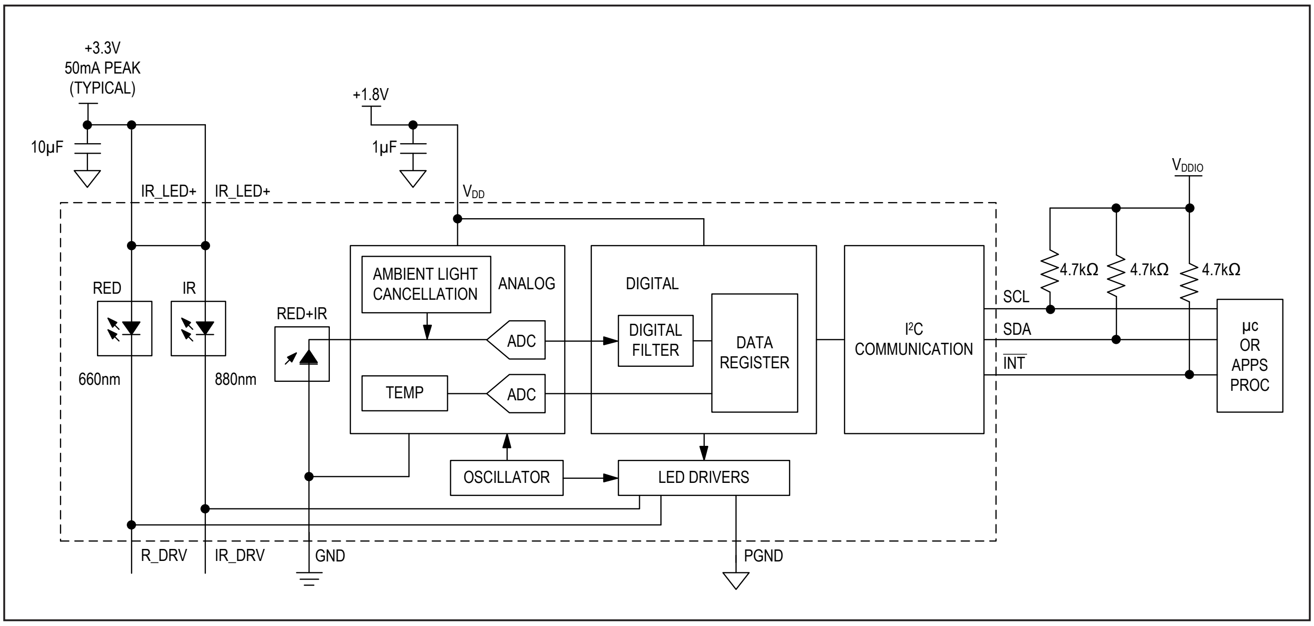


Figure 10. Reading Multiple Bytes of Data from the MAX30100

Typical Application Circuit



Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX30100EFD+	-40°C to +85°C	14 OESIP (0.8mm pitch)

+Denotes a lead(Pb)-free/RoHS-compliant package.

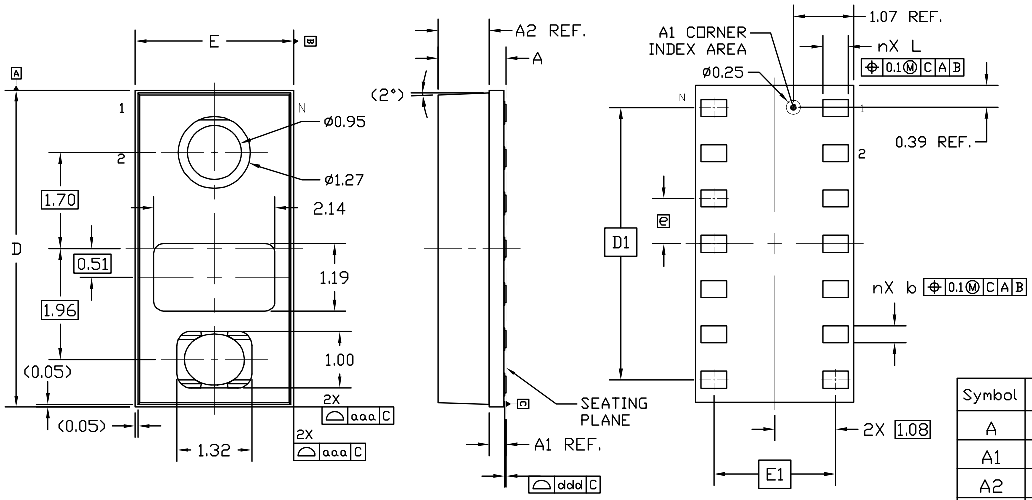
Chip Information

PROCESS: BiCMOS

Package Information

For the latest package outline information and land patterns (footprints), go to **www.maximintegrated.com/packages**. Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

PACKAGE TYPE	PACKAGE CODE	OUTLINE NO.	LAND PATTERN NO.
14 OESIP	F142D5+2	21-0880	90-0461



TOP VIEW SIDE VIEW BOTTOM VIEW

Symbol	MIN	NOM	MAX
A	1.1	1.2	1.3
A1		0.3 REF.	
A2		0.9 REF.	
b	0.25	0.30	0.35
D	5.5	5.6	5.7
D1		4.8 BSC	
E	2.7	2.8	2.9
E1		2.15 BSC	
e		0.8 BSC	
L	0.40	0.45	0.50
N		14	
aaa		0.1	
ddd		0.08	

NOTES:

1. ALL DIMENSIONS ARE IN MILLIMETERS. ANGLES ARE IN DEGREES.
2. DIMENSIONING & TOLERANCING CONFORM TO ASME Y14.5M-1994.
3. N IS THE TOTAL NUMBER OF TERMINALS.
4. MATERIAL MUST BE COMPLIANT WITH MAXIM SPECIFICATION 10-0131 FOR SUBSTANCE CONTENT, MUST BE Eu ROHS COMPLIANT WITHOUT EXEMPTION AND PB-FREE.
5. ALL DIMENSIONS APPLY TO Pb FREE (+) PARTS.
6. PACKAGE CODE: F142D5+2

-DRAWING NOT TO SCALE-

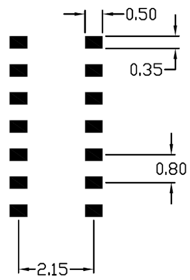
maxim integrated

TITLE: PACKAGE OUTLINE, 14L OLGA, 2.8x5.6x1.2mm

APPROVAL	DOCUMENT CONTROL NO.	REV.	
	21-0880	B	1/1

Package Information (continued)

For the latest package outline information and land patterns (footprints), go to **www.maximintegrated.com/packages**. Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.



NOTES:
1. REFERENCE PKG. OUTLINE: 21-0880.
2. LAND PATTERN COMPLIES TO: IPC7351A.
3. TOLERANCE: +/- 0.02 MM.
4. ALL DIMENSIONS APPLY TO PbFREE (+) PKG. CODE ONLY
5. ALL DIMENSIONS IN MM.

-DRAWING NOT TO SCALE-

maxim integrated™

This document (including dimensions, notes & specs) is a recommendaton based on typical circuit board manufacturing parameters. Since land pattern design depend on many factors unknown to Maxim (eg. user's board manufacturing specs), user must determine suitability for use.
This document is subject to change without notice.
Contact technical support at http://www.maxim-ic.com/support for further questions.

TITLE: PACKAGE LAND PATTERN, [F142D5+2] OLGA

APPROVAL	DOCUMENT CONTROL NO.	REV.	
	90-0461	A	1/1

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	9/14	Initial release	—

Department of Electrical and Electronics Engineering
Vidya Academy of Science & Technology
Thalakkottukara, Thrissur - 680 501
(http://www.vidyaacademy.ac.in)