Smart Clock BLE

Project application using PSoC 63 BLE

About this document

# Scope and purpose

The Smart Clock is an intelligent alarm clock that can be controlled by voice or hand. It not only displays the time but also functions as a virtual assistant with features like playing music and controlling other devices. The objective of this project is to develop adaptive smart technologies, including improved user experience, multiple command options, and the ability to detect air quality. The project timeline includes various milestones such as testing the sensors, integrating components, and final testing.

# Intended audience

Student of Electronics Engineering, Embedded System Engineering, and Application Engineer.

Table of contents

[About this document 1](#_Toc176443645)

[Table of contents 1](#_Toc176443646)

[1 Introduction 2](#_Toc176443647)

[2 Design and Application 3](#_Toc176443648)

[2.1 FreeRTOS 3](#_Toc176443649)

[2.2 Software 3](#_Toc176443650)

[2.2.1 ModusToolboxTM 3](#_Toc176443651)

[2.2.2 Cyberon Dspotter Modeling Tool V2 4](#_Toc176443652)

[2.3 Hardware 5](#_Toc176443653)

[2.3.1 CYBLE-416045-02 6](#_Toc176443654)

[2.3.2 MEMS Microphone 9](#_Toc176443655)

[2.3.3 BME680 9](#_Toc176443656)

[2.3.4 LCD Oled 128x64 I2C 9](#_Toc176443657)

[2.3.5 MAX98357A I2S 3W Class D Amplifier 10](#_Toc176443658)

[2.3.5.1 Heading 4 10](#_Toc176443659)

[References 12](#_Toc176443660)

[Revision history 13](#_Toc176443661)

[Disclaimer 14](#_Toc176443662)

# Introduction

Smart clock is an intelligent device that can be operated in three different modes, namely button (manual), voice command, and BLE modes. With flexible operational capabilities, this smart clock can meet the needs of users in setting the date and time, setting daily and monthly alarms, and detecting surrounding environmental conditions such as temperature, pressure, humidity, and gas. In button (manual) mode, users can easily set the date and time using the buttons available on the device. This mode is especially useful for users who prefer to set the device manually or do not have access to voice technology. Using the button mode, users can set the date and time quickly and easily and can choose from the various alarm options available to wake them up at the right time. In voice command mode, users can set the date and time, set alarms, and get information about the surrounding environment using only voice commands. This mode is especially useful for users who want to manage the device hands-free or have limited mobility. By using voice command mode, users can manage the device more easily and quickly, and can request information about the surrounding environment without having to touch the device. In BLE (Bluetooth Low Energy) mode, users can connect the smart clock with other devices such as smartphones or tablets to set the date and time, set alarms, and get information about neighborhood conditions. This mode is especially useful for users who want to integrate the smart clock with other devices they use daily. By using BLE mode, users can manage the device more easily and quickly, and can monitor the surrounding environment more effectively. With these features, this smart clock can help users manage their schedules and monitor the surrounding environment more effectively and efficiently. In addition, this smart clock can also help users raise awareness of the importance of managing time and monitoring environmental conditions, thereby improving their overall quality of life.

In addition, this smart clock is also equipped with various other features, such as the ability to detect changes in temperature, humidity, and air pressure, as well as the ability to monitor air quality and detect the presence of harmful gases. With these features, this smart clock can help users monitor their surroundings more effectively and efficiently, thereby improving their overall quality of life. In the long run, this smart clock can be a very useful device for users in managing their schedules and monitoring environmental conditions. With flexible operational capabilities and advanced features, this smart clock can help users raise awareness of the importance of managing time and monitoring environmental conditions, thereby improving their overall quality of life.

A white box with a screen and buttons

Description automatically generated

Figure 1 Smart Clock

# Design and Application

## FreeRTOS



FreeRTOS is a class of RTOS that is designed to be small enough to run on a microcontroller - although its use is not limited to microcontroller applications.

A microcontroller is a small and resource constrained processor that incorporates, on a single chip, the processor itself, read only memory (ROM or Flash) to hold the program to be executed, and the random access memory (RAM) needed by the programs it executes. Typically the program is executed directly from the read only memory.

Microcontrollers are used in deeply embedded applications (those applications where you never actually see the processors themselves, or the software they are running) that normally have a very specific and dedicated job to do. The size constraints, and dedicated end application nature, rarely warrant the use of a full RTOS implementation - or indeed make the use of a full RTOS implementation possible. FreeRTOS therefore provides the core real time scheduling functionality, inter-task communication, timing and synchronisation primitives only. This means it is more accurately described as a real time kernel, or real time executive. Additional functionality, such as a command console interface, or networking stacks, can then be included with add-on components [1].

## Software

There is several software used to complete theSmart Clock, namely Modus Toolbox, EAGLE, Autodesk Inventor, Cyberon Dspotter Modeling Tool V2, and Tera Term. The following is an explanation of the software used:

### ModusToolboxTM

|  |
| --- |
|  |

Figure 2 ModusToobox Software

ModusToolbox is used to develop applications that run on Infineon microcontrollers. ModusToolbox supports a wide range of Infineon microcontroller devices, including PSoC™ Arm® Cortex® Microcontrollers, TRAVEO™ T2G Arm® Cortex® Microcontrollers, XMC™ Industrial Microcontrollers, AIROC™ Wi-Fi devices, AIROC™ Bluetooth® devices, and USB-C Power Delivery Microcontrollers. Embedded software assets for ModusToolbox™ include board support package (BSP), peripheral drivers, middleware, and code samples. BSPs are provided for a variety of Infineon devices, including Arm® Cortex® PSoC™ Microcontrollers, XMC™ Industrial Microcontrollers, AIROC™ Wi-Fi devices, Bluetooth™ devices, and USB-C Power Delivery Microcontrollers. ModusToolbox supports several IDEs, namely:

* Eclipse IDE for ModusToolbox™
* Microsoft Visual Studio Code

There are two APIs that can be used in ModusToolbox to program the PSoCTM, namely Hardware Abstarction Layer (HAL) and Pripheral Driver Library (PDL). PDL is a low-level device-specific library that reduces the need to understand the use of registers and bit structures, thus facilitating software development for a wide range of peripherals in PSoC™ devices. HAL is a non-device-specific high-level library that provides a common interface for peripheral configuration[2].

### Cyberon Dspotter Modeling Tool V2

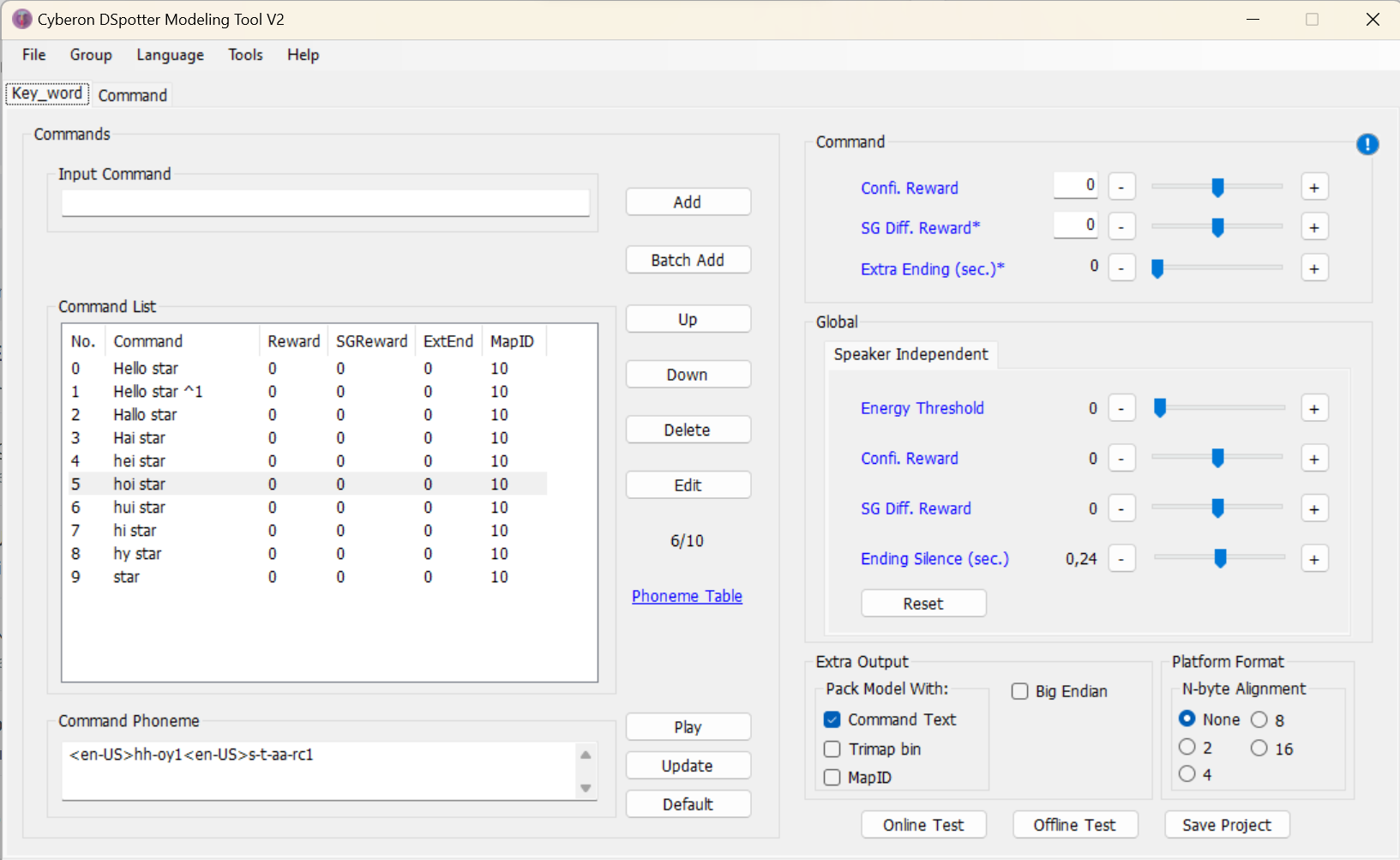
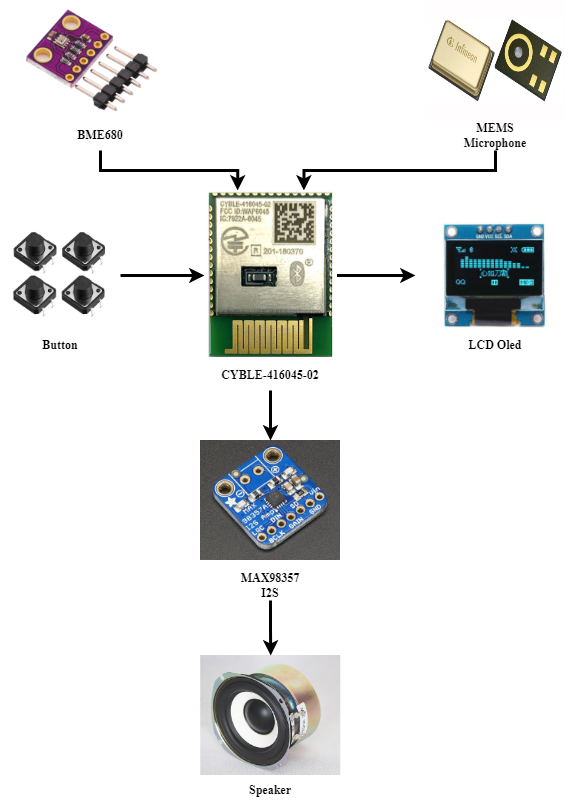


Figure 3

Cyberon Dspotter Modeling Tool V2 is an application used to create voice commands that are integrated with the cyberon library for PSoCTM. To use the Cyberon DSpotter Modeling Tool V2 application, first have an account so that you can configure the commands that will be used. In creating a custom voice command model using Cyberon DSpotter Modeling Tool V2, you can choose several languages to use for voice triggers. To create a voice command, you must first model the voice command used as keywords such as hello google and command words such as (Light on, Light off) using the Cyberon DSpotter Modeling Tool V2 application. After creating and saving the modeling, then input the *[project\_name]\_pack\_withTxt.bin* file into the data folder in the voice command program workspace and copy the file name into the cyberon\_data.s file in the data folder in the project.

Figure 4 Cyberon DSpotter Modeling Tool V2 software

## Hardware



**Figure 5**

The following hardware is used in Smart Clock, there are several components used including CYBLE-416045-02, BME680, MEMS Microphone, 128x64 pixel Oled LCD, MAX98357A I2S 3W Class D Amplifier, and CH340. The following is an explanation of the components used:

### CYBLE-416045-02

CYBLE-416045-02 or PSoC™ 63 MCU with Bluetooth® LE connectivity, commonly called PSoC™ 6 Bluetooth® LE, is an ultra-low power PSoC™ device A low power PSoC™ device specifically designed for Bluetooth Low Energy devices.



**Figure 6**

The PSoC™ 6 Bluetooth® LE device is a programmable embedded system-on-chip that integrates the following in a single chip:

* Dual-CPU microcontroller: CM4 and CM0+
* Bluetooth® LE 5.2 subsystem
* Programmable analog and digital peripherals
* Up to 1 MB flash and 288 KB SRAM
* Fourth generation CAPSENSE™ technology.

The CYBLE-416045-02 has a CPU Subsystem consisting of two Arm Cortex cores and associated bus and memory: M4 with FPU and MPU, and M0+ with MPU. The M4 and M0+ cores have an 8-KB instruction cache (I-Cache) with four-way associativity. The subsystem also includes independent DMA controllers with 32 channels each, a cryptographic accelerator block, 1 MB on-chip Flash, 288 KB SRAM, and 128 KB ROM.

The CBLE-416045-02 has two DMA controllers, which support CPU independent access to memory and peripherals. Each DMA has 16 channels each.

The CYBLE-416045-02 has 1 MB flash with an additional 32K flash that can be used for EEPROM emulation for longer storage and a separate 32 KB flash block that can be securely locked and can only be accessed via an immutable keyword (OTP). There is a 288 KB 32-KB Detailed Retention SRAM, which can be fully retained or retained in user-defined 32-KB block increments. There is a 128 KB SROM that contains boot and configuration. This ROM will guarantee Secure Boot if user flash authentication is required. The eFuse 1024-bit OTP memory can provide a unique and immutable identifier on each chip. This immutable key can be used to access secure flash.

CYBLE-416045-02's power system provides assurance that the appropriate voltage level is required for each respective mode and will delay mode entry (on power-on reset (POR), for example) until the voltage level matches that required for proper function or generate a reset (brown-out detect (BOD)) when the power supply drops below the specified voltage level. The VDDD supply (1.7 to 3.6 V) powers the in-chip buck regulator or low-dropout regulator (LDO), which can be selected by the user. In addition, both the buck and LDO offer a choice of (0.9 or 1.1 V) core operating voltage (VCCD). This choice allows the user to choose between two system power modes:

* The Low Power (LP) system operates VCCD at 1.1 V and offers high high and offers high performance, with no restrictions on device configuration.
* Ultra Low Power (ULP) system operates VCCD at 0.9 V for exceptionally low power but imposes restrictions on clock speed.

The CYBLE-416045-02 clock system is responsible for providing clock to all subsystems that require clock and for switching between different clock sources without interruption. In addition, the clock system ensures that no metastable conditions occur. The clock system for the CYBLE-416045-02 consists of an internal main oscillator (IMO) and an internal low-speed oscillator (ILO), an external crystal oscillator (ECO) and WCO, a PLL, a frequency locked loop (FLL), and provisions for an external clock. The FLL will provide fast startup at high clock speeds without waiting for a PLL lockup event (which can take up to 50 µs). The clock can be buffered and brought out to a pin on the Smart I/O port. The 32-kHz oscillator can be trimmed to within 2 ppm using a higher accuracy clock. ECO will provide ±20 ppm accuracy and will use an external crystal.

CYBLE-416045-02 combines a Bluetooth Smart subsystem containing PHY and Link Layer (LL) engines with an embedded security engine. The physical layer consists of a digital PHY and RF transceiver that transmits and receives GFSK packets at a rate of 2 Mbps over the 2.4 GHz ISM band, which is compliant with Bluetooth Smart Bluetooth Specification 5.0. The baseband controller is a composite hardware and firmware implementation that supports both master and slave modes. Key protocol elements, such as Host controller Interface (HCI) and link control, are implemented in the firmware. Critical functional blocks, such as encryption, CRC, data whitening, and access code correlation, are implemented in hardware (in the LL engine).

The RF transceiver contains an integrated balun, which provides a single-ended RF port pin to drive a 50-ohm antenna through a matching/filtering network. In the receive direction, this block converts the RF signal from the antenna into a digital bit stream after performing GFSK demodulation. In the transmit direction, it performs GFSK modulation and then converts the digital baseband signal into radio frequency before transmitting it into the air through the antenna.

The main features of BLESS are as follows:

* Master and Slave single-mode protocol stack with logic link protocol and adaptation protocol (L2CAP), attribute (ATT), and security manager (SM) protocols.
* API access to generic attribute profile (GATT), generic access profile (GAP), and L2CAP.
* L2CAP connection-oriented channel (Bluetooth 4.1 feature).
* GAP feature
* Role Broadcaster, Observer, Peripheral, and Central
* Security mode 1: Level 1, 2, and 3
* User-defined advertising data
* Supports multiple bonds
* GATT Features
* GATT client and server.
* Supports GATT sub-procedures.
* 32-bit universally unique identifier (UUID) (Bluetooth 4.1 feature).
* Security Manager (SM)
* Pairing methods: Just works, Passkey Entry, and Out of Band.
* LE Secure Connection Pairing model.
* Authenticated man-in-the-middle (MITM) protection and data signing.
* LL
* Master and slave roles.
* 128-bit AES engine.
* Low-duty cycle advertising.
* LE Ping.
* Supports all SIG-adopted BLE profiles.
* Power levels for advertisement (1.28s, 32 bytes, 0 dBm) and Connection (300 ms, 0 bytes, 0 dBm) are 42 µW and 70 µW respectively.

The CYBLE-416045-02 has five SCBs (Serial Communication Blocks), each of which can implement I2C, UART, or SPI Interfaces. Two SCBs (SCB\_6 and SCB\_8) share the same pin connections and cannot be used simultaneously. One of these SCBs (SCB\_8) will operate in Deep Sleep mode with an external clock, this SCB will only operate in Slave mode (requiring an external clock).

The hardware I2C block implements a full multimaster and Slave Interface (capable of multimaster arbitration). The block can operate at speeds up to 1 Mbps (Fast Mode plus) and has flexible buffering options to reduce interrupt overhead and CPU latency. The block also supports EZI2C which creates an address range in CYBLE-416045-02 memory and effectively reduces I2C communication for reading from and writing to arrays in memory. In addition, the block supports 256 byte-deep FIFOs for receiving and sending, by increasing the time allotted for the CPU to read data, greatly reducing the need for clock stretching caused by the CPU not reading data in time. FIFO mode is available on all channels and is especially useful in the absence of DMA. I2C peripherals are compatible with Standard-mode, Fast-mode, and Fast-Mode Plus I2C devices as defined in the NXP I2C bus specification and user guide (UM10204). The I/O of the I2C bus is implemented with GPIOs in open-channel mode.

A full-featured UART that operates up to 8 Mbps. This mode supports automotive single wire interface (LIN), infrared interface (IrDA), and SmartCard (ISO7816) protocols, which are all minor variants of the basic UART protocol. In addition, it supports a 9-bit multiprocessor mode that allows addressing of connected peripherals through common Rx and Tx lines. Common UART functions such as parity error, pause detection, and frame error are also supported. A 256-byte FIFO allows much greater CPU service latency to be tolerated.

SPI mode supports full Motorola SPI, TI Secure Simple Pairing (SSP) (basically adding an initial pulse used to synchronize the SPI Codec), and National Microwire (a half-duplex form of SPI). The SPI block supports EZSPI mode where data exchange is reduced to reading and writing arrays in memory. The SPI interface operates with a 25-MHz clock.

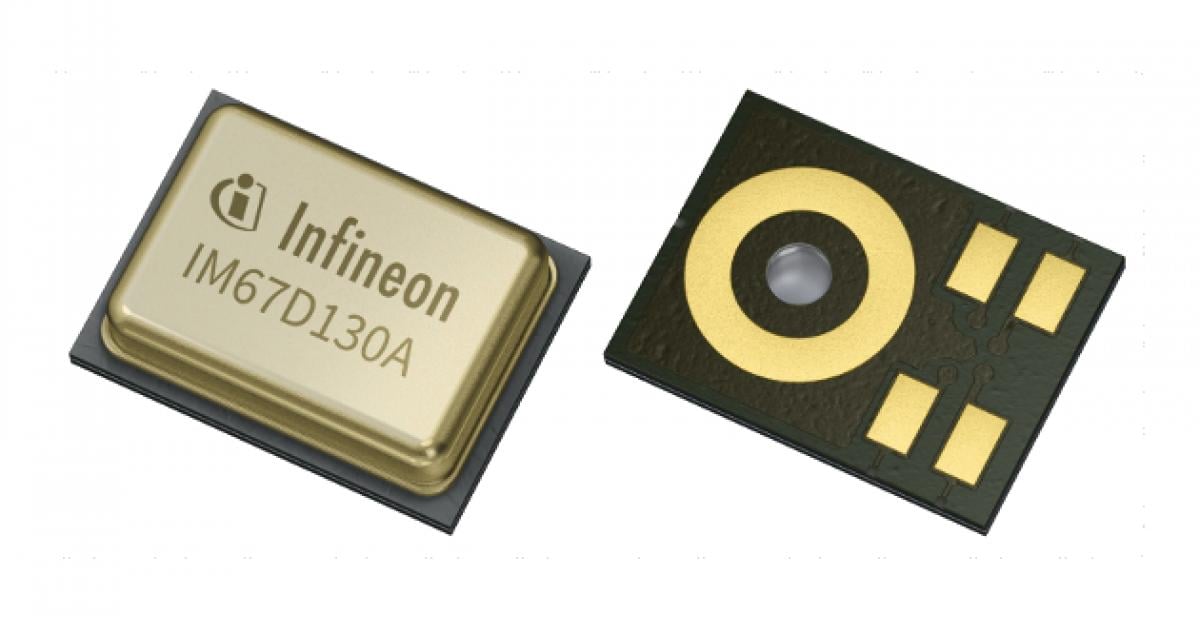
The CYBLE-416045-02 has up to 36 GPIOs, which implement the following:

* Eight drive strength modes:
* Analog input mode (input and output buffers are disabled)
* Input only
* Weak pull-up with strong pull-down
* Strong pull-up with weak pull-down
* Open drain with strong pull-down
* Open drain with strong pull-up
* Strong pull-up with strong pull-down
* Weak pull-up with weak pull-down
* Input threshold selection (CMOS or LVTTL)
* Hold mode to lock the previous state (used to maintain the I/O state in system hibernation mode)
* Can select slew rate for dV/dt related noice control to improve EMI

CYBLE-416045-02 also has an audio subsystem, this subsystem consists of the following hardware blocks:

This subsystem consists of an I2S block and two PDM channels. The PDM channel connects to the bit stream output of the PDM microphone. The PDM processing channels provide drop correction and can operate at clock rates ranging from 384 kHz to 3.072 MHz and generate word lengths of 16 to 24 bits at audio sample rates of up to 48 ksps. The I2S interface supports master and slave modes with Word Clock rates up to 192 ksps (8-bit to 32-bit words)[3].

### MEMS Microphone



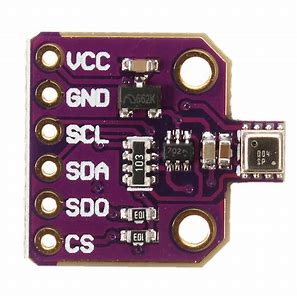
**Figure 7 MEMS Microphone**

XENSIV™ MEMS microphones feature very low self-noise (high SNR), very low distortion (THD) even at high sound pressure levels (SPL), very tight phase matching and sensitivity, flat frequency response with low LFRO (low frequency roll-off), as well as very low group delay. Combined with its selectable power mode and extremely small package size, the Infineon XENSIV™ MEMS microphone is ideally suited for consumer electronics with excellent audio capture functions and also for certain industrial applications such as predictive maintenance and security. IM69D130 is a high performance digital MEMS microphone making use of Infineon’s Dual Backplate MEMS technology to deliver 105dB dynamic range and high output linearity up to 130dBSPL. The application benefits are crystal clear audio signals, extended pick-up distance and sensitivity to both soft and loud signals - from whispered speech to rock concerts[4].

**Summary of Features**

* 69 dB(A) signal-to-noise ratio
* Below 1 percent distortions at 128 dBSPL (AOP - 130 dBSPL)
* Digital (PDM) interface with 6 μs group delay at 1 kHz
* Tight sensitivity (-36 ±1 dB) and phase (± 2 deg) tolerances
* 28 Hz low frequency roll-off

### BME680



**Figure 8 BME680 Sensor with I2C and SPI interface**

The BME680 sensor is an absolute barometric pressure sensor specifically designed for mobile applications such as cell phones, GPS modules, or watches. It is based on Bosch's proven piezo-resistive pressure sensor technology, which features high accuracy and linearity as well as long-term stability and high EMC resistance. The sensor has small dimensions and low power consumption, making it possible to be implemented in battery-powered devices[5].

### LCD Oled 128x64 I2C

Ultra-low power OLED displays are used for visual interaction in small form factors. The OLED has a 0.96-inch display with 128x64 resolution and a 4-pin I2C interface based on the SSD1306 controller. It is easy to connect to a microcontroller with the I2C interface. The display has a very low power consumption of 0.04W during standard operation, as it does not require a backlight. It supports a wide operating voltage range between 3.3V and 5V.



**Figure 9 Oled LCD 128x64 I2C**

### MAX98357A I2S 3W Class D Amplifier



**Figure 10 MAX98357A I2S 3W Class D Amplifier**

The MAX98357A is a 3W Class D audio amplifier integrated circuit (IC) designed for use in various audio applications. The MAX98357A is an I2S (Inter-IC Sound) amplifier, which means that it receives digital audio data via the I2S interface and converts it into an analog audio signal. This signal is then amplified to drive speakers or other audio output devices. As a Class D amplifier, the MAX98357A operates with high efficiency, typically above 80%. This is achieved by using a switching amplifier topology, which reduces power and heat losses compared to traditional Class AB amplifiers. The MAX98357A is capable of delivering up to 3W of power to speakers, making it suitable for a wide range of applications, including portable audio devices, smart speakers, and other compact audio systems[6]. Some key features of the MAX98357A include:

* I2S digital audio input
* 3W output power
* High efficiency (typically above 80%)
* Low quiescent current (IQ) for low power consumption
* Integrated thermal protection and over-current protection
* Small package size (e.g., WLP or TQFN)

## Programming Flow

First, we need to start with PSOC system initialization and FreeRTOS configuration. After that, we can create tasks that are required to perform various functions such as reading sensors, handling buttons, recognizing voice commands, updating displays, sending notifications, and handling BLE communication. Each task has a different priority and period. The priority determines the order of task execution, while the period determines the time between executions of the same task. For example, the task to read the sensor has a high priority and a period of 10ms, while the task to update the display has a low priority and a period of 100ms. Once the tasks are created, we can implement the code for each task. For example, the task to read the sensor will read the sensor value and store it in a global variable, while the task to handle the button will handle the button press and send the signal to other tasks. Once all the tasks are implemented, we can integrate them using the xTaskCreate function to create the task and vTaskStartScheduler to start the scheduler. The scheduler will run the tasks concurrently and organize the execution order of the tasks based on priority and period. Thus, the PSOC system can run multiple tasks simultaneously and improve the efficiency of using system resources. FreeRTOS allows us to develop more complex and efficient systems by utilizing features such as multitasking, priority, and period.

A diagram of a computer program

Description automatically generated

**Figure 11 Flowchar main.c smart clock program**

### Button task

Button task bertanggungjawab untuk membaca button yang di tekan oleh user dengan beberapa pilihan yaitu shoert press, double click, and long press.

A diagram of a flowchart

Description automatically generated

### DisplayOled task

## Electronics and Hardware design

#### Heading 4

// some code

0110 abcdefghi

Code listing 2 Title

|  |
| --- |
| 1. Code in code table |

References

1. <https://www.freertos.org/>

1. <https://www.infineon.com/cms/en/design-support/tools/sdk/modustoolbox-software/>
2. <https://www.infineon.com/dgdl/Infineon-CYBLE-416045-02-DataSheet-v02_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0ee63f136e57>
3. <https://www.infineon.com/cms/en/product/sensor/mems-microphones/mems-microphones-for-consumer/im69d130/>
4. <https://www.bosch-sensortec.com/products/environmental-sensors/gas-sensors/bme680/#documents>
5. <https://learn.adafruit.com/adafruit-max98357-i2s-class-d-mono-amp/overview>

Revision history

| Document revision | Date | Description of changes |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Disclaimer