Requirements Definition

MRD – Market Requirements Document (**CRD** – Customer Requirements Document) – includes nontechnical items.

Functional Requirements – for ES they come from the product requirements (what the product must do and under what conditions).

See the attached guide to MRDs and Specs, UserReqsEngSpecs.PDF.

This is a very important phase of the final-product requirements: sometimes it is done with “overkill” - by specifying some features which are desirable - “just in case”.

We are going to skip the job of Marketing engineers here. Let’s try to formulate some functional requirements for our Embedded System.

Functional Requirements

Our ES must be made on a single PCB and include/provide:

• modern low power and high speed microcontroller with Flash and SRAM memories enough to create a medium complexity system;

• wireless connectivity via several technologies (Wi-Fi, Bluetooth, Sub-1GHz);

• stereo recording of audio signals;

• temperature measurements;

• humidity measurements;

Functional Requirements, Cont.

- measurements of atmospheric pressure;
- measurements of magnetic field in 3 (X, Y, Z) directions;
- measurements of acceleration in 3 directions;
- 2 pushbuttons (Reset and User’s);
- 3 LEDs for indication of user defined states;
- on-board debugger/programmer with SWD Interface; ;
- connectors for external modules;
- fast non volatile memory for data recording.

Other Requirements

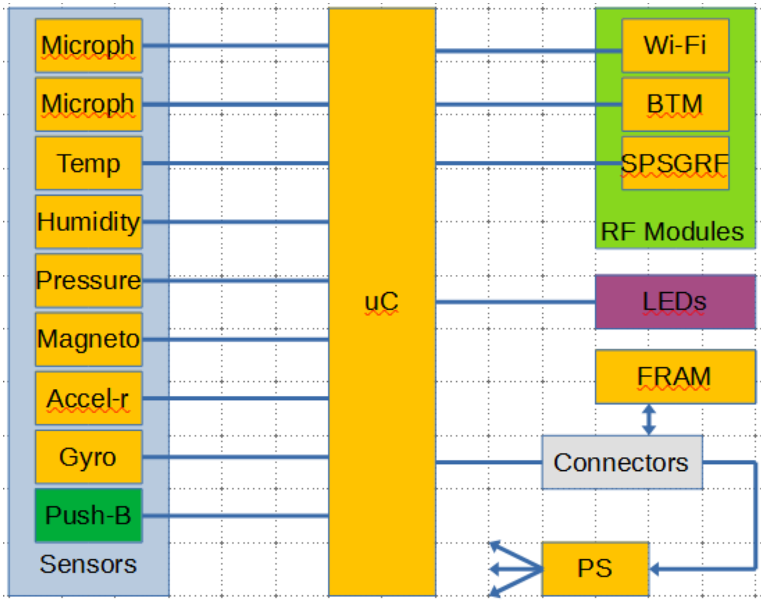
- Single PCBA Main ES plus a detachable FRAM Module.
- Power must be supplied from ST LINK USB or external Power Supplies. Selection of the source to be made by jumper.
- Most of the active components on the PCBA should work from 3V3 nominal voltage.
- Main PCBA must have connectors for Arduino-standard modules, for PMOD-standard, and for SWD/JTAG connections.
- No requirements for an enclosure or firmware at this time.

Architecture of Embedded Systems

Tammy Noergaard, the Author of the book Embedded Systems Architecture, said: “...defining and understanding the architecture of an embedded system is an essential component of good system design”.

Let’s try to draw an architecture of our ES, based on the inputs given in previous slides. Because we deal with electrical HW design now, we do not consider ME (Mechanical Engineering), FW and SW requirements for the ES.

Detailed Architecture of ES



Creating a Block Diagram

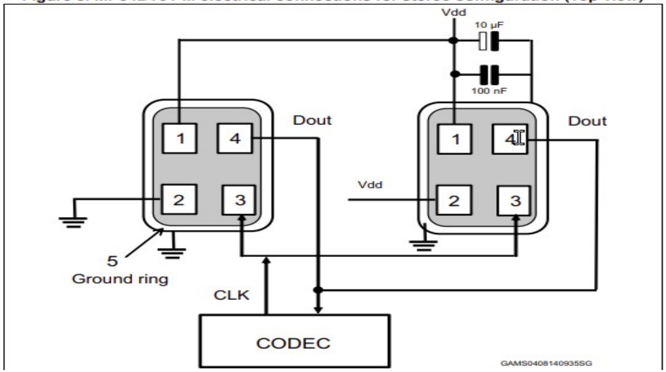
We already have the necessary modules for our ES, now it is time to define the interconnections between them – to create a Block Diagram. I will show you the way to do it on the example of two microphones. The rest is your Assignment 2. (For sensors select I2C bus – it takes less connections with uC than SPI. RF Modules need higher speed bus, so use SPI bus.)

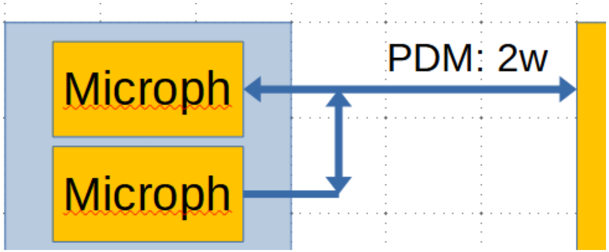
Download and read the data sheet for SPK0641HT4H-1-Rev-A mike. Decide how you are going to connect microphones to the uC. They use PDM (Pulse Density Modulation) output, so we are looking for uC inputs accepting PDM. They go to Digital Filter for Sigma-Delta Modulators (DFSDM) block in the uC .Put the name of the bus (PDM) and total number of connection lines (wires) on the bus going to uC, as shown on the next slide.

Digital Microphones with PDM Interface

How Microphones are connected on a reference schematic from data sheet and on our Block Diagram:

Figure 5. MP34DT01-M electrical connections for stereo configuration (Top view)





Here we have CLK and Dout lines common for both microphones - total 2 lines of the PDM bus. Dout Left is activated on low CLK, Right – on high.

Partial Engineering Specifications

- Microphones must have SNR >60dB, <-20dBFS sensitivity;
- Temp sensor must be +/-0.5°C accurate in 15 to 45C range;
- Humidity accuracy: ± 3.5% rH in 20 to +80% rH range;
- Barometer for 260-1260 hPa range, error less than 1hPa;
- Magnetometer for +/- 4 Gauss range;
- Accelerometer for +/- 2G range;

Other modules' and PCB's specs will be considered later.

Schematic Design Tools

To design an embedded system (or any electronic and electrical), one has to use some Computer Aided Design (CAD) tools, such as schematic capture and analysis, PCB layout and analysis, mechanical design tools, and Integratged Development Environment (IDE) tools for FW and SW development.

We are going to use KiCAD, Ver.6.01, a free software suite for Electronic Design Automation (EDA).

It integrates schematic capture, simulation, and PCB layout design. KiCAD's tools help to create a bill of materials, artwork, Gerber files, and 3D views of the PCB and its components.

KiCAD Download and Installation

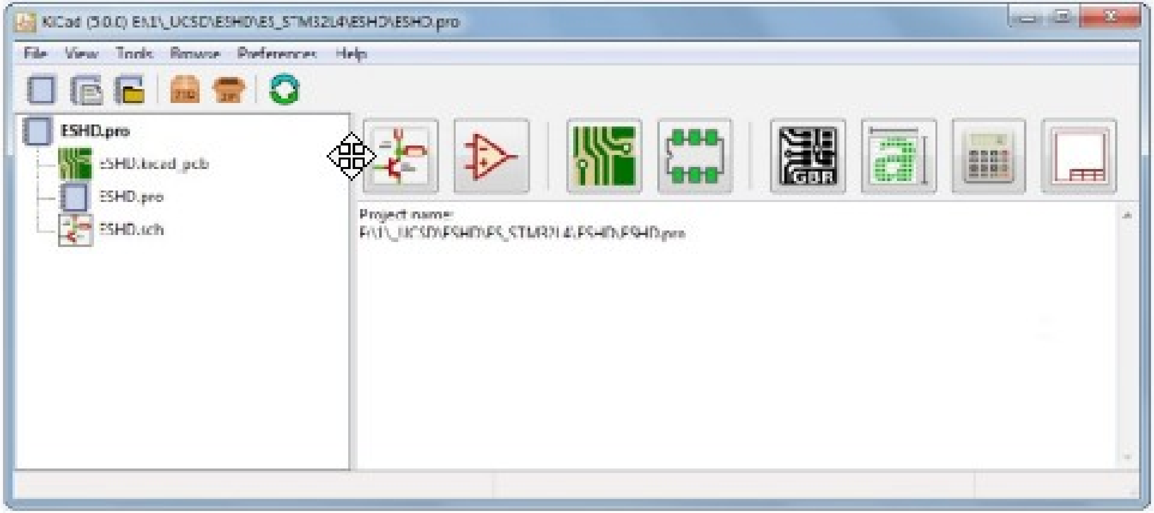
Go to the web site http://kicad-pcb.org/ and download the latest stable version in accordance with your operating system.

Then go to https://docs.kicad.org/#_getting_started , and download the first four guides in PDF: Introduction, Getting Started, KiCAD, and Schematic Editor. Install KiCAD on your computer.

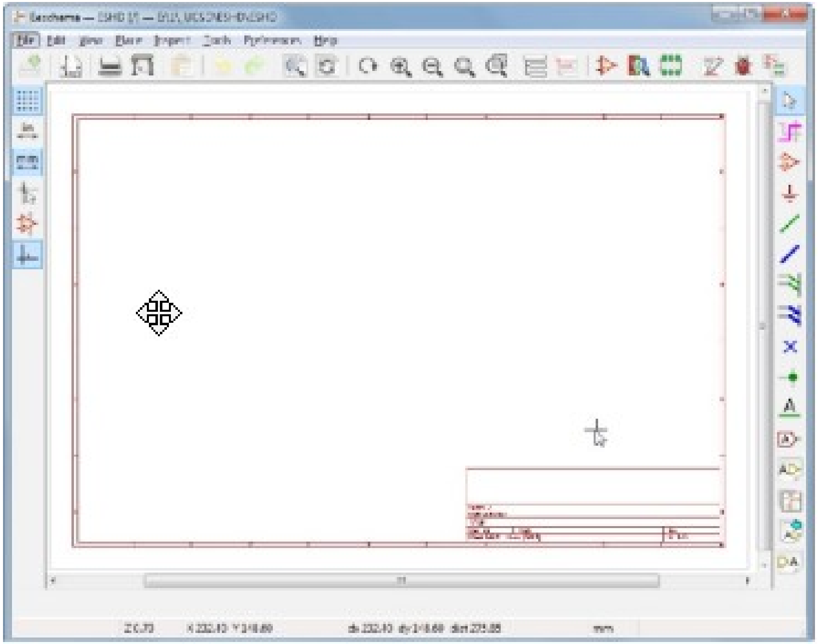
KiCAD is one of a few well conceived packages of SW: it creates a schematic from symbols, then designer assigns the footprints to symbols, creating components. KiCAD also has the Design Rule Check (DRC), a BOM and a netlist generator in the schematic SW called Eeschema.

KiCAD Project Window

Click on the KiCAD icon on your desktop to open a project:



Eeschema, Schematic Editor



Hierarcical Schematic

Change grid to Imperial: menu → View → Units, select Inches. Change Grid Settings (menu → Grid Properties) to 0.1”.

Click on menu → Add Sheet, Look at the highlighted icon on right side tool bar.

Move cursor to the position X=1.600, Y=2.000 (look at the bottom raw of the window) and left click and release. Draw a rectangle about 1.2x2.4”. Click and release to complete the rectangle. In the pop-up window write a File name (P2) and a Sheet name (P2). Create 4 more blocks 1.2x2.4” by (copy & paste) to the right of 1st one. Press Esc key to end drawing mode. Hover cursor over block, press “e” - to edit the names of each of five sheets (see next slide).

Multipage Schematic

Take a look at the Title Block of the main page, notice Id: 1/6.

Name the blocks (pages) on the main page (P1):

P2_uController (Id: 2/6) – for Microcontroller

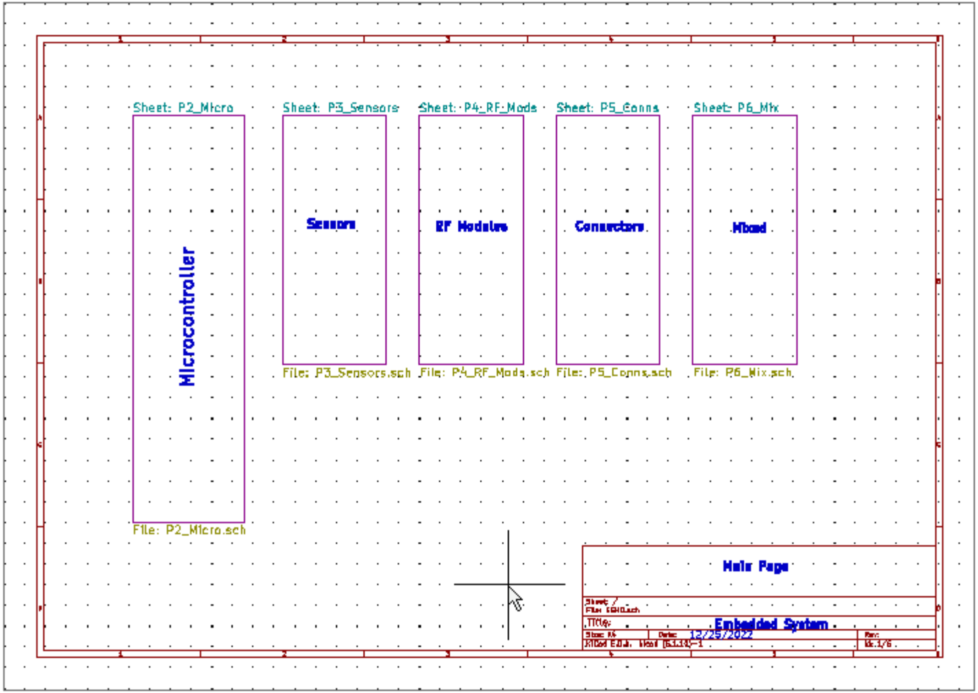
P3_Sensors (Id: 3/6) – for sensors (including microphones)

P4_RF_Modules (Id:4/6) - for Wi-Fi, SPSGFR, and Bluetooth modules

P5_Connectors (Id: 5/6) – for Arduino, JTAG, and PMOD connectors

P6_Mix: (Id: 6/6) Here you will place power supplies, pushbuttons and LEDs.

6-page Schematic



Opening and Closing Pages

On the top schematic page right click on a block and select Enter Sheet - to open it;

When on the block page, right click on the open space and select Leave Sheet – to close it.

KiCAD has a list of “Hotkeys”, many of them are intuitive: R – rotate, X -mirror on X, Y – mirror on Y, E – edit, M – move, etc. The list can be found in the Eeschema and other manuals recommended before on Slide 11.

I attached a file Hotkeys.pdf to Lesson 2. It was good for 5.12 version of KiCAD.

Placing Symbols

Open P2 – uC. From the Menu go to Place → Add Symbol and click on schematic sheet or just press “a”.

Select the library MCU_ST_STM32L4, doble click on it and go down to select STM32L475VGTx microcontroller and double click on it.

Place the microcontroller in the middle of the left half of the schematic page and place other symbols: resistors, capacitors, crystals, etc - on this page. For reference look at the sheets 2 and 3 of the STM IoT kit.

If you do not find a symbol in the libraries, you will have to create it or substitute the part. You must store each symbol inside a library file, so let’s create a new library, MySymLib, before you start creating the symbol.

Symbol Editor

From Project window go to Tools → Symbol Editor.

In the Symbol Editor click on menu File → New Library.

Select add your library to Project libraries. Name the new library "MySymLib_XX.lib". It will be placed in the same subdirectory as your project. (..ESHD/Project_XX, where XX – your initials). Click Save. You will see your library on the left side among other KiCAD libraries. To make our life easier, let's create a symbol of microphones (U2, U10 in IoT schematic) from a prototype, SPH0641LU4H-1, which can be found in Sensor_Audio library.

Modifying Symbol

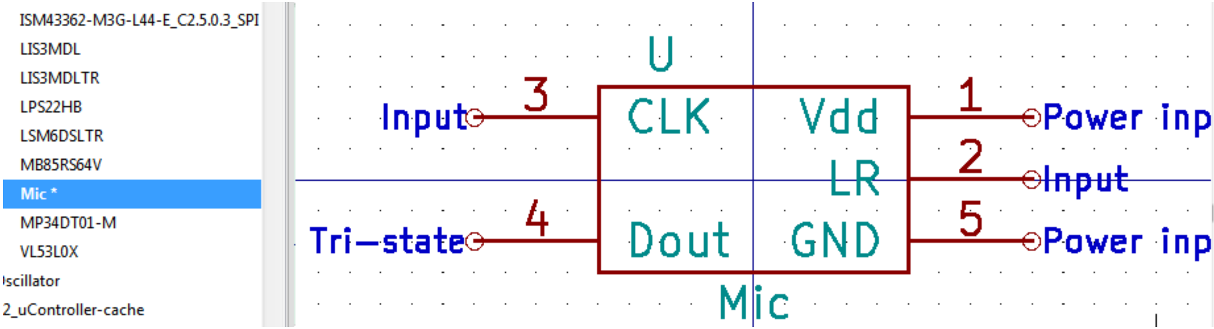
In the Symbol Editor go to Sensor_Audio library and double click on it. Select SPH0641LU4H-1 and double click on it. After that, Save (it) As, find your MySymLib_XX and save. Now it is in your library, and you can make modifications.

Double click on each pin of the symbol and make it looking as on the next slide: Pins 1 and 5 ares Power Inputs, pins 2 and 3 are Inputs, pin 4 is Output.

See an example of the Microphone on the next slide.

Use schematic and BOM of SM32L475 Kit for references to other components. Read their data sheets for more info.

Digital Microphone Symbol (U2, U10)



Look at the schematic of SM32L475 Kit and assign proper numbers and names to the pins. Notice a difference with the data sheet: pin4 is tri-state – not output. I explained the reason in comments to slide 8: when CLK is low – left mike's output is active and right output is in high-Z state. When CLK is high – both change state. Save it.

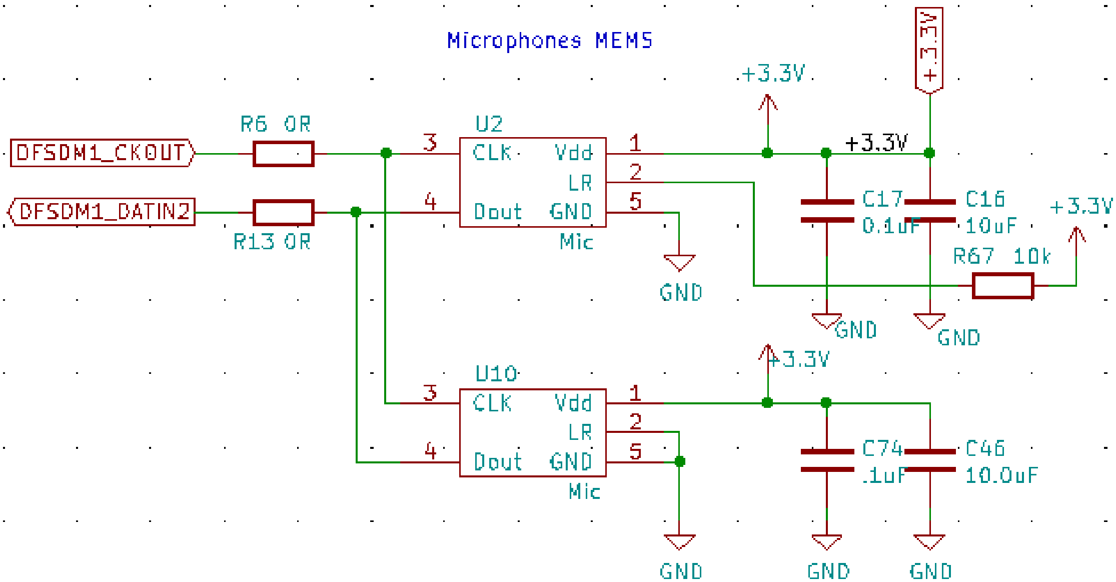
Symbols for Other Devices

The same way as above you can create the symbols for other devices, if you cannot find them in the libraries. In some cases, it is easier to modify an existing symbol and save it under different name. Also, use **snapeda.com, digikey.com, mouser.com** - for symbols and footprints.

Symbols for resistors, capacitors, diodes, crystals, transistors, ICs, and connectors can be found in their libraries too.

Look at the schematic of SM32L475 Kit and place the symbols on each page accordingly, with all necessary accessories (GND to Power filtering capacitors, etc).

Fragment of Page 3, Sensors



Making Electrical Connections

To place a “wire” (connection), move the pointer to the pin of the component and press “w”. Now drag the wire to the next pin or wire, left clicking at each corner of turn (most of the turns are 90 degrees).

Place “Global Labels” with the same names at both ends of the wires going to another page(s).

Here is a short and simple tutorial from schematic to layout:

https://www.youtube.com/watch?v=l9b_6WLemmg

Later we will continue with the schematic; that’s all for now.