# Embedded Product Design:<br>Design paper

Magnus Mortensen, Nicolas Munafo - Shanghai Polytechnic University, 2019/12

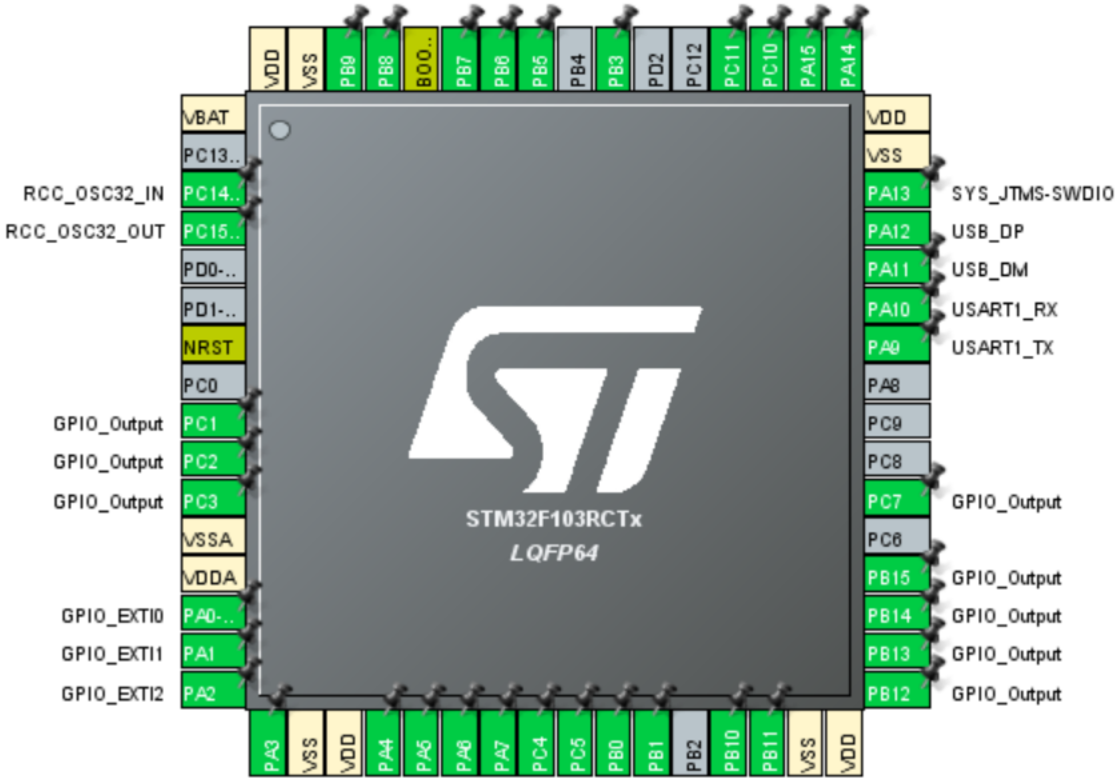
In the class *Embedded Software Development,* the main project was to create a smart clock using a board giving in the beginning of the semester. Alongside developing the smart clock, every fundamental aspect designing and developing embedded system is being taught, which should be reflected in project. In this paper our design thoughts for this project will be gone through and reasoned for.

In the first part of the of the paper the configuration of the STM32F103RCTx CPU will be explained.

## Configuration of CPU

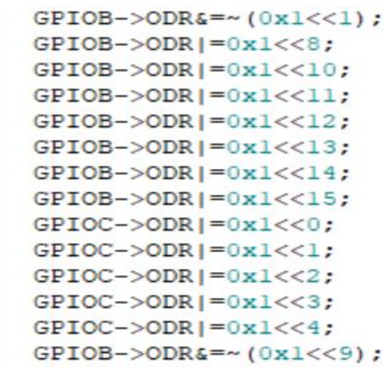
The first part was configuring a virtual CPU identical to the one on our board, namely a STM32F103RCTx. For this we used the software STM32CubeMX. We went through the CPU ports and designated each with the function we desired. For this we referred to the board schematics provided to us by the professor. Keeping the final requirements in mind we finetuned each port's parameters. It is very essential to the performance of the CPU that every port is assigned it function according to the board schematics. If a CPU pin get assigned with the wrong function, then it can course serious performance issues.

When the CPU was configured, we used STM32CubeMX to generate the template code which would eventually run on the actual board, the operating system. The CubeMX is powerful and important tool in our design process as it was used for this project. With CubeMX we found our CPU model and configured it as seen below. This was of course done on basis of the schematics of the CPU. Once all the CPU pins essential to our needs was configured in the CubeMX software, we generated the code using CubeMX generate code tool. This tool therefore makes the process of configuring a CPU much more accessible. After the code has been generated, then the project is basically ready to take form. To write the code that would run on our board we switched software from CubeMX to Kiel uVersion 5. This tool is meant for writing C++ code which is what the compatible with the board. The next section will therefore go into more detail on how we wrote this code and what is was designed to do.



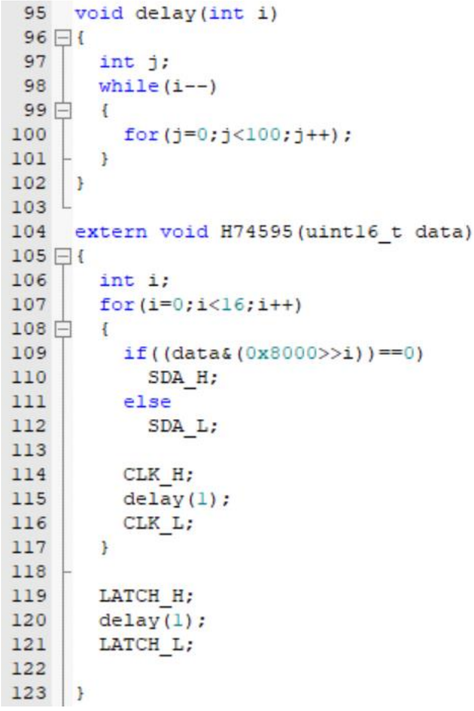
## Implementation of the design

Once in Keil uVision 5 we use the previously generated code. Adding on top of this we started to write additional code to implement or desired functions. We want to control the LED panel to display the time. The passing of each second must be calculated precisely, which is why we don't want to depend on software level computations as this might vary on different computers. We implement CPU system clock feature to precisely time one second. After we have a reliable time calculation, we start writing the code to display a digital clock, 24-hour system. Depending on the incoming data the clock will be adjusted to keep in sync.



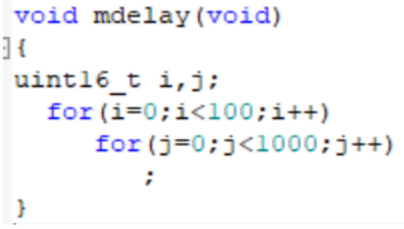
```
GPIOB->ODR&=~(0x1<<1);
GPIOB->ODR|=0x1<<8;
GPIOB->ODR|=0x1<<10;
GPIOB->ODR|=0x1<<11;
GPIOB->ODR|=0x1<<12;
GPIOB->ODR|=0x1<<13;
GPIOB->ODR|=0x1<<14;
GPIOB->ODR|=0x1<<15;
GPIOC->ODR|=0x1<<0;
GPIOC->ODR|=0x1<<1;
GPIOC->ODR|=0x1<<2;
GPIOC->ODR|=0x1<<3;
GPIOC->ODR|=0x1<<4;
GPIOB->ODR&=~(0x1<<9);
```

This connection with the internet will make sure that, no matter what time zone the clock finds itself in, the system time will be correct. That is not the only point of the incoming data.



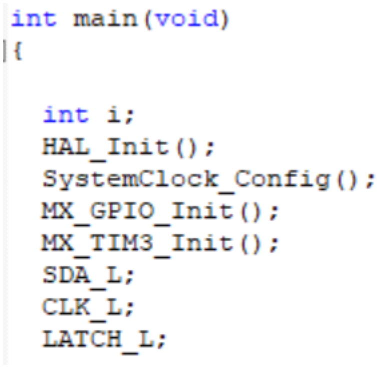
```
void delay(int i)
{
  int j;
  while(i--)
  {
    for(j=0;j<100;j++);
  }
}

extern void H74595(uint16_t data)
{
  int i;
  for(i=0;i<16;i++)
  {
    if((data&(0x8000>>i))==0)
      SDA_H;
    else
      SDA_L;

    CLK_H;
    delay(1);
    CLK_L;
  }

  LATCH_H;
  delay(1);
  LATCH_L;

}
```

On the board itself are stored around 100 mp3 files which can be used as alarms. The users can set the desired times when they want a file to play. These files are organized in folders, and if so desired every file in a folder can be played at a certain time.



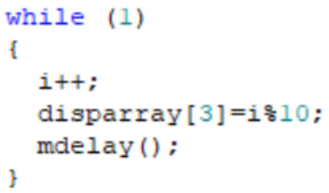
```
void mdelay(void)
{
uint16_t i,j;
   for(i=0;i<100;i++)
       for(j=0;j<1000;j++)
           ;
}
```

Part of the data is how long the alarm should play. Thus, we program the necessary code to interpret the incoming json code and deduce from that what files the user wants activated at what specific time, how long it should play and what days this needs to be repeated. There is also a speaker attached to the board. This speaker will play the current weather forecast at specific times, or when the user presses the necessary button. This weather forecast is also pulled from the internet and ergo current information.



```
int main(void)
{

  int i;
  HAL_Init();
  SystemClock_Config();
  MX_GPIO_Init();
  MX_TIM3_Init();
  SDA_L;
  CLK_L;
  LATCH_L;
```

Future plans are to include a Bluetooth chip, so that the smart clock can be used as a Bluetooth speaker which may be brought to remote locations. This implementation would also facilitate the communication with the android device that currently connects with the smart clock. To connect the two devices the user currently needs to press a button on the smart clock to initiate connection. With Bluetooth the user would only need to connect once, and afterwards the will be able to send alarm information at any time with no additional effort.



```
while (1)
{
  i++;
  disparray[3]=i%10;
  mdelay();
}
```

## Conclusion

Due to the fact that the members of the project group both where new to embedded software design, the startup of the project went slower and therefore also impacted the final result of the project. Would we start a new project to day the result would undoubtedly be of a better quality. To elaborate on this project accomplished, it gave both members a much deeper understanding on how to design embedded system, especially on the intricate part of configuration a CPU.

The project overall did however have some missing functionality when it comes to the more extensive features such as connection with the server or being able to play music. This was primally due to the fact that no group member was experienced in the language C++. There where therefore a lot time spent researching fundamental and syntax problems, which would have been spent on these features instead. The project did on the other hand fulfill all basic and core functionality, which has been a great gateway into the world of designing embedded system. Therefore, both members views the overall project as an success, due to the fact that is has introduced all fundamental aspects of embedded software design.