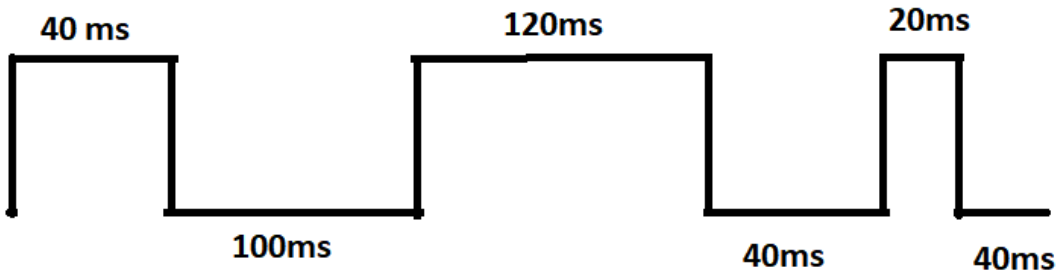
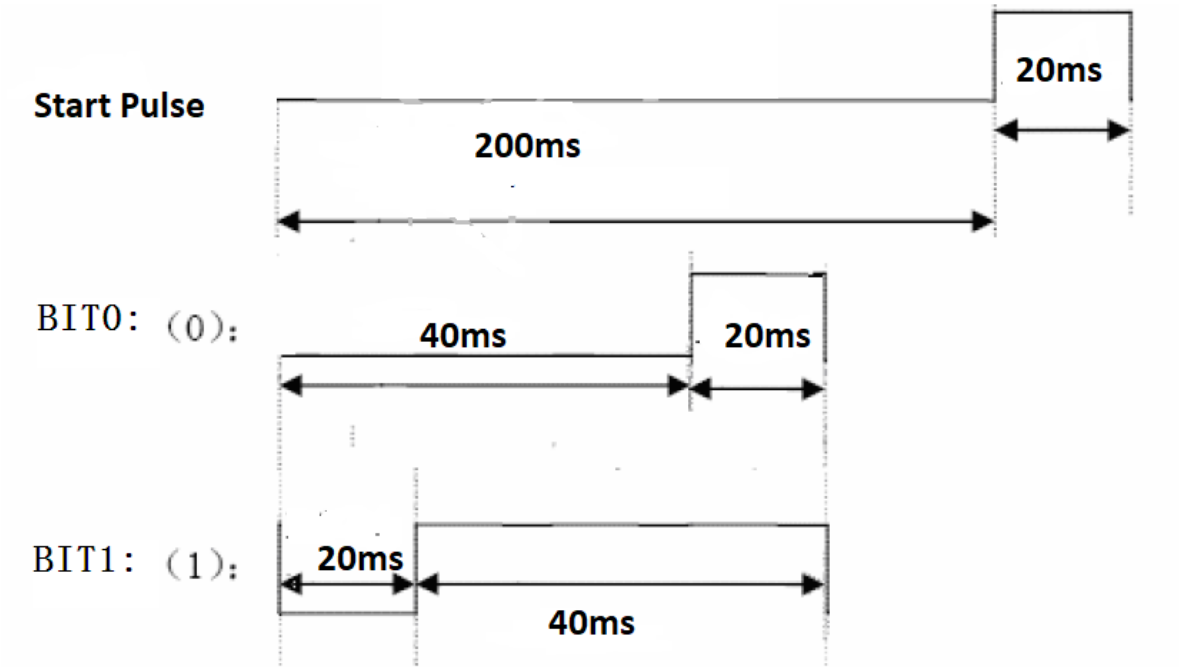
Task: Develop a single project for below tasks

1. Turn On motor "X" after every 10minuts. We need to send below pulse for turning motor "X" ON for once.



2. On digital GPIO, we will receive 55 bits data with below pulse. Before sending data (55bits), each time start pulse will be sent. After receiving 55 bits, convert data (55 bits) to bytes and send over UART. Using pulse timing we can differentiate bit 0 and 1. Data is in MSB to LSB order



3) Sensor adxl375 connected over SPI and we need to read sensor accelerometer data every 100ms. Someone wrote function code as below. Can we make it better? We also need to send data over UART.

```c
#define ADX375_DATAX0_READ    0x32

void adxlRead(uint8_t *data, uint8_t length) {
    uint8_t add = ADX375_DATAX0_READ;
    uint8_t readData[length];
    add |= 0x80;  // read operation

    HAL_SPI_Transmit(&ADXL_COMM_SPI, &add, 1, 100);
    HAL_SPI_Receive(&ADXL_COMM_SPI, &readData[0], 1, 100);
    add++;
    HAL_SPI_Transmit(&ADXL_COMM_SPI, &add, 1, 100);
    HAL_SPI_Receive(&ADXL_COMM_SPI, &readData[1], 1, 100);
    add++;
    HAL_SPI_Transmit(&ADXL_COMM_SPI, &add, 1, 100);
    HAL_SPI_Receive(&ADXL_COMM_SPI, &readData[2], 1, 100);
    add++;
    HAL_SPI_Transmit(&ADXL_COMM_SPI, &add, 1, 100);
    HAL_SPI_Receive(&ADXL_COMM_SPI, &readData[3], 1, 100);
    add++;
    HAL_SPI_Transmit(&ADXL_COMM_SPI, &add, 1, 100);
    HAL_SPI_Receive(&ADXL_COMM_SPI, &readData[4], 1, 100);
    add++;
    HAL_SPI_Transmit(&ADXL_COMM_SPI, &add, 1, 100);
    HAL_SPI_Receive(&ADXL_COMM_SPI, &readData[5], 1, 100);
    add++;

    memcpy(data, readData, length);
}
```

Rules:

- Use any STM32 microcontroller.
- Use any one UART port for all 3 tasks.
- Use any one digital pin for motor “X” pulse.
- Don't use any RTOS.