

Directives

To make you get familiar with the PIC32 programming, we have provided you with an example of LED rotation code from which you can learn the dynamics of the PIC32 programming.

1. Open mikroC PRO for PIC32 application and create a new project.
2. Set Device Clock to 80 MHz and device name P32MX795F512L.
3. Copy the below example code into <ProjectName>.c file that is automatically created for you and save the file.
4. Build the project.
5. Make sure that you connected the ports as shown in the Figure 1. Note that some ports of PIC32 might not work consider using other ports. However, if you used for example PORTC instead of PORTD, change the given code accordingly because in the code it is assumed to you used PORTD.
6. Connect your board to computer and open mikroBootloader USB HID v2.10 that is located on the desktop on lab computers.
7. Click the PP_RST button on the PIC32 daughter board and press connect on the mikroBootloader. Note that after pressing button you only have a few seconds to connect. Then, browse the hex file that is created where your project locates. Finally, press Begin Uploading. After few seconds, your code will be uploaded and run.

Notes

1. mikroC does not have a type called boolean hence, consider using integer for that purpose.
2. Make sure that jumpers are set to POW 3V3 if possible.
3. Top right port which is not visible in the Figure 1 is PORT A.
4. Stop button that is used in the example code is push-button on the Beti Board whose label is 1.

Example Code

```
// This code shows and rotates the pattern (10001000) right or stops based on the
input coming from the user. The pattern is to be shown on the LEDs.
int stop = 0;
int initial = 0b01110111; //Initial pattern. Note that 0 means on, while 1 means off.
int right = 1;
void main(){
    TRISD = 0x0; // All bits of PORTD are output. ~0 means output~
    // Three bits of PORTA are inputs but only one of them is used in this example as a
    stop button, others are redundant. ~1 means input~
    TRISA = 0b111;
    // From PORTD, outputs will be sent to LEDs. Make sure that you physically connected
    them by looking at the Figure 1, in the directives document.
    // Initial pattern is sent to the LEDs through PORTD.
    PORTD = initial;
    while(1){
        int lsb;
        int mask;
        // Stop button is the push-button which is labeled as 1 on the board.
        if(PORTABits.RA1 == 0){ // If stop button clicked
            stop = !stop;
            if(!stop){
                // If process restarted, copy initial pattern into PORTD.
                PORTD = initial;
            }
        }
        if(!stop){
            //Rotate right
            lsb = PORTD & 0x1; // Extract least significant bit
            mask = lsb << 7; // Least significant bit will be the msb of the
shifted pattern
            PORTD = (PORTD >> 1) | mask; // Paste lsb to the leftmost bit the
right shifted portd
        } else {
            //Do not shift anything, that is, stop.
            PORTD = 0b11111111;
        }
        delay_ms(1000); // Wait 1 second.
    }
}

// Rotation ends here
```

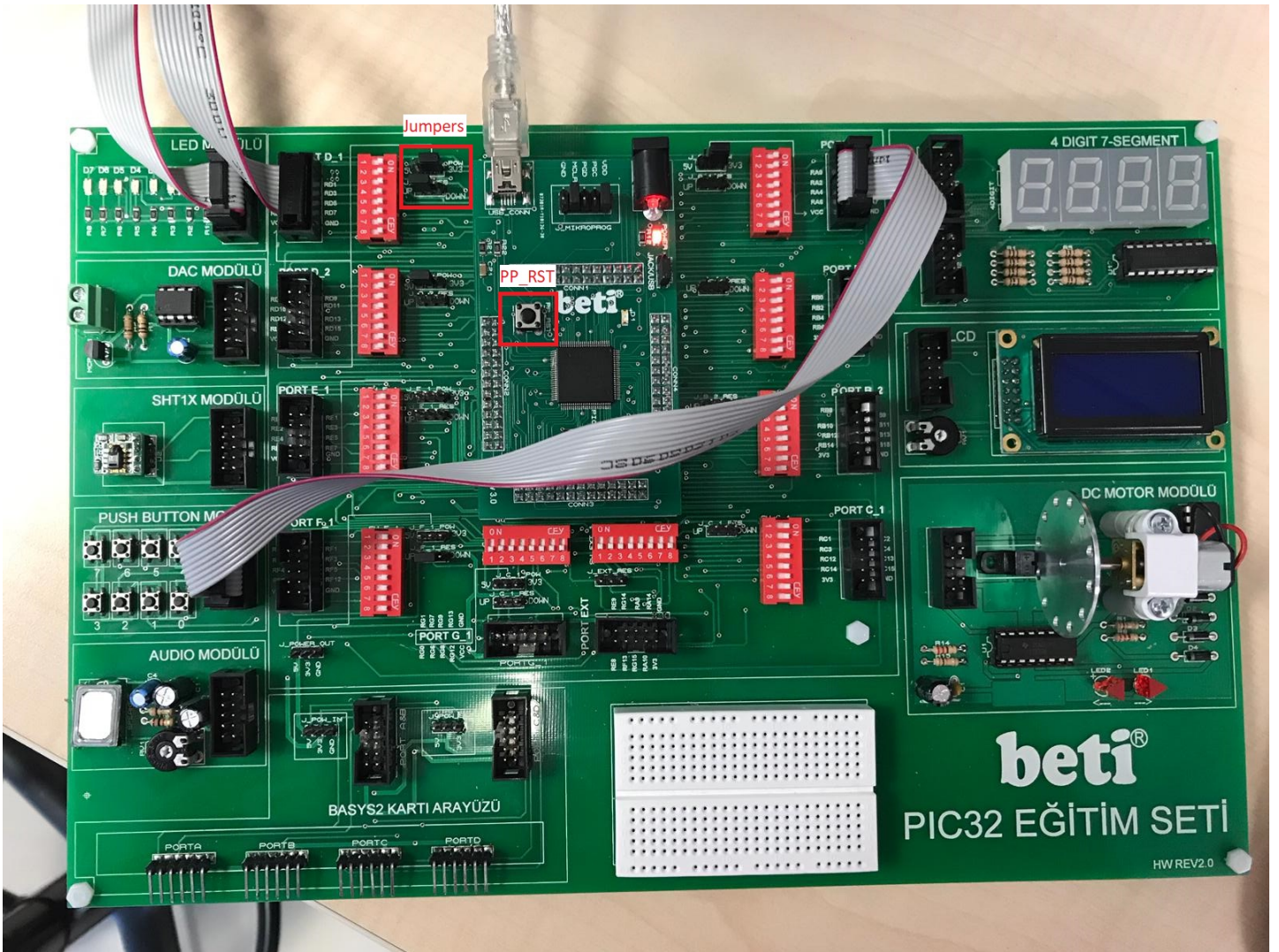


Figure 1: Beti Board configuration for the example code