

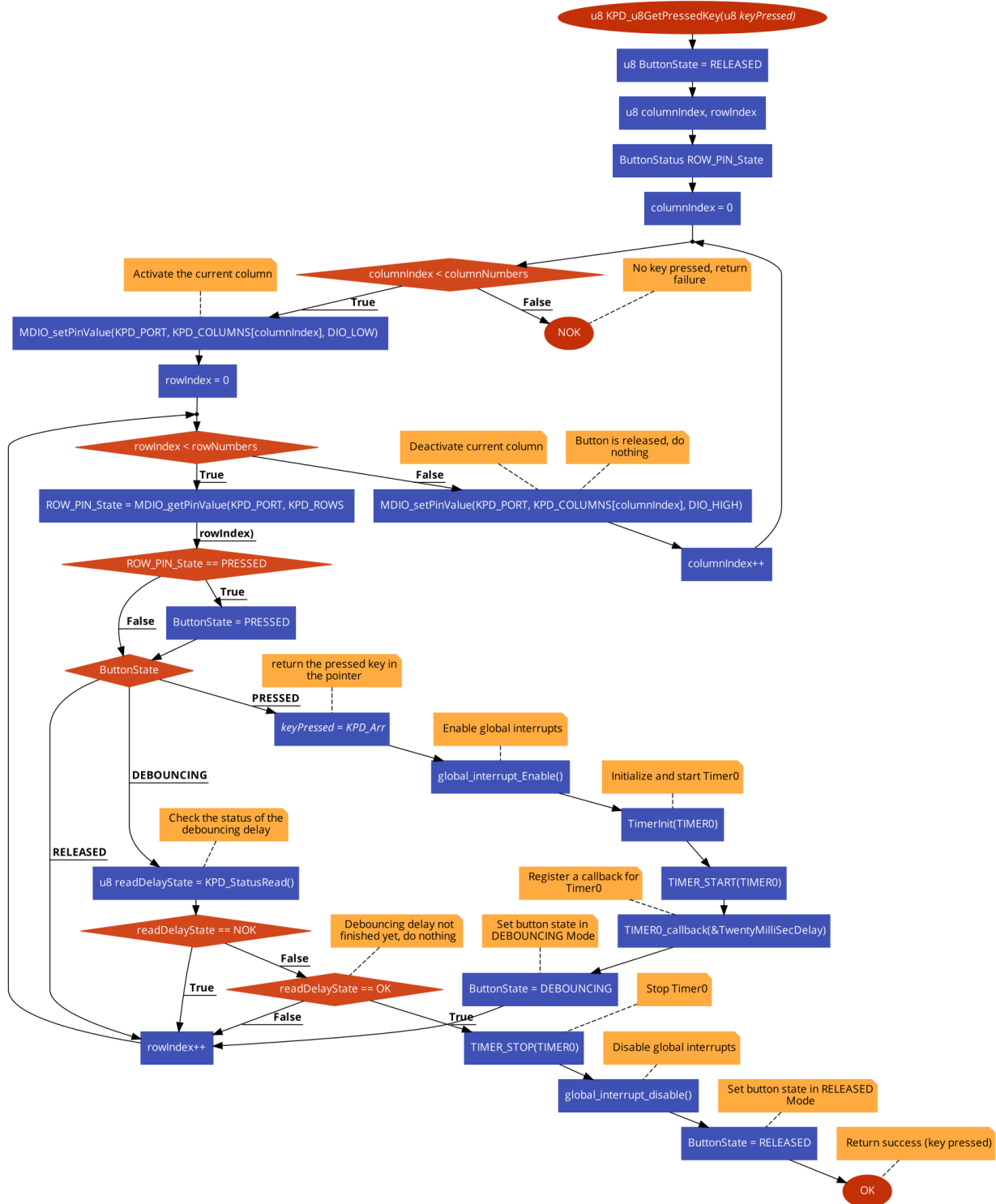
# KP & LCD Non-blocking

## 1. Non-blocking Keypad :

- Pseudocode :

```
Non-blocking Keypad :
Initialize Keypad and Flags
getKeyState(*keyPressed) function to be non-blocking 3*3
{
    check column index
    Set ButtonState to RELEASED //Initial state for button as unpressed
    //Iterate to active each column and check the row's value
    for each column in Column numbers:
        Activate Column //Set low
        //Iterate to check the row's value
        for each row in Row numbers:
            if Button is Pressed(row):
                Set ButtonState to PRESSED
                switch case on -> button states
                case Button is Pressed:
                    Store Pressed Key in the passed argu (poiner) //Passed the pressed key
                    Enable the global interrupt
                    Initialize Timer0
                    Start Timer 0
                    Set Timer Callback function for a 20ms delay
                    Set ButtonState to DEBOUNCING
                    case Button is DEBOUNCING:
                        if function delay Finished :
                            Stop Timer 0
                            Disable Global Interrupt
                            Set ButtonState to RELEASED
                            return KEY_OK
                    Deactivate current column //set it to High
                    return KEY_NOK
}
```

- Flowchart :



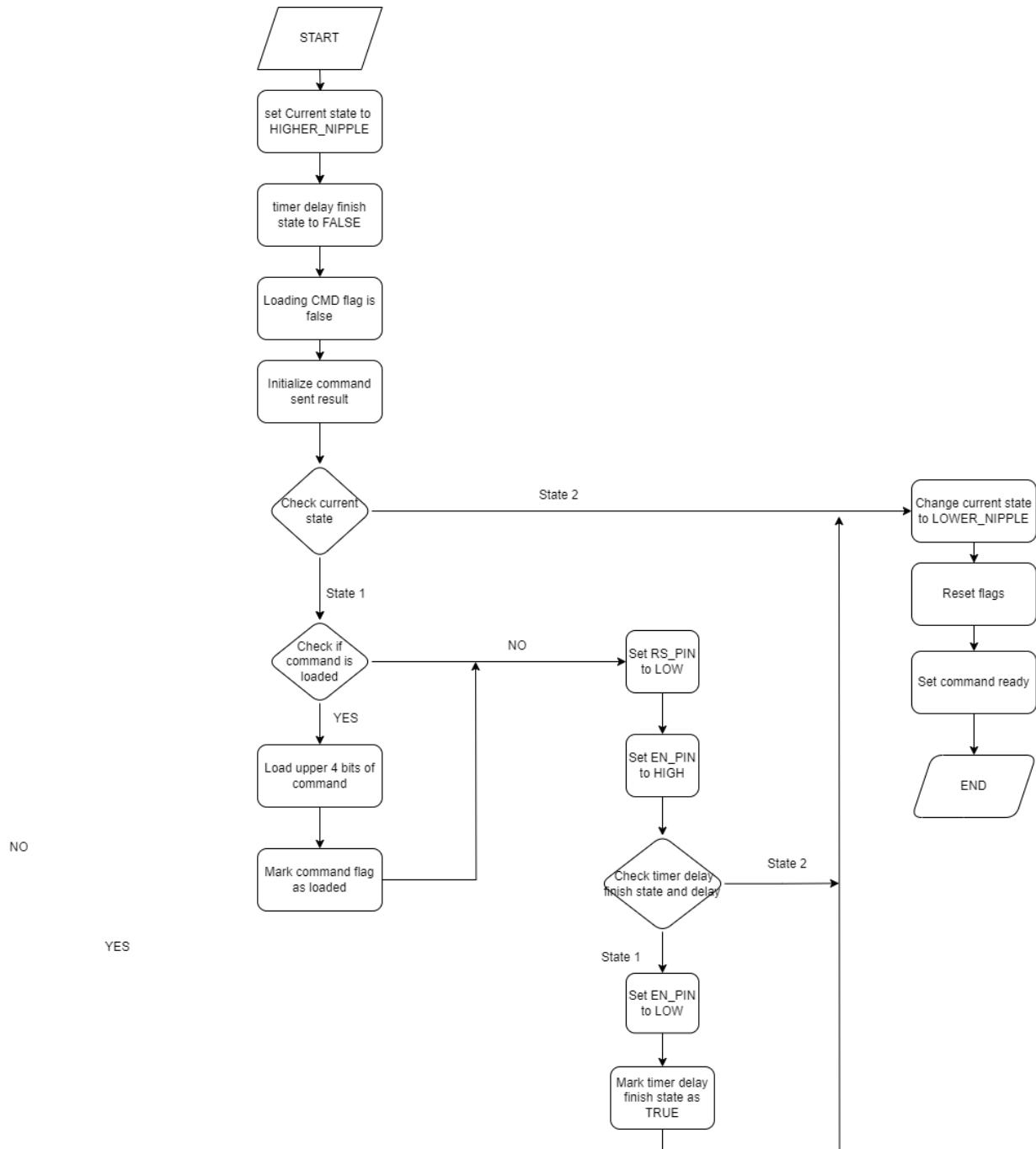
## 1. Non-blocking LCD:

### ☐ **LCD\_send\_command :**

- Pseudocode :

```
FUNCTION LCD_send_command(u8 u8_a_command):
Set the LCD initial CMD state to LOADING_HIGHER_NIPPLE_CMD_STATE
Set a timer delay finish state to FALSE
Set a flag indicating that the command is not loaded to FALSE
Initialize a flag for the command sent result
SWITCH the current state DO
CASE LOADING_HIGHER_NIPPLE_CMD_STATE:
Set the flag for the command being sent to NOT READY
IF the command is not loaded:
Set the upper 4 bits of DATA_REGISTER to the upper 4 bits of u8_a_command
Mark the command as loaded
Set RS_PIN of the LCD command port to LOW
Set EN_PIN of the LCD command port to HIGH
IF the timer delay finish state is FALSE and a delay of 10 ms has passed:
Mark the timer delay finish state as TRUE
Set EN_PIN of the LCD command port to LOW
ELSE IF the timer delay finish state is TRUE and a delay of 10 ms has passed:
Change the current state to LOADING_LOWER_NIPPLE_CMD_STATE
Reset the timer delay finish state and command loaded flags
END IF
CASE LOADING_LOWER_NIPPLE_CMD_STATE:
IF the command is not loaded:
Set the lower 4 bits of DATA_REGISTER to the lower 4 bits of u8_a_command
Mark the command as loaded
Set EN_PIN of the LCD command port to HIGH
IF the timer delay finish state is FALSE and a delay of 10 ms has passed:
Mark the timer delay finish state as TRUE
Set EN_PIN of the LCD command port to LOW
ELSE IF the timer delay finish state is TRUE and a delay of 10 ms has passed:
Change the current state to LOADING_HIGHER_NIPPLE_DATA_STATE
Reset the timer delay finish state and command loaded flags
Set the flag for the command being sent to READY
END IF
END CASE
END SWITCH
RETURN the flag indicating the command sent result
END FUNCTION
```

- Flowchart :



☐ **LCD\_send\_character :**

- Pseudocode :

```
FUNCTION HLCD_sendChar(u8 u8_data):
DECLARE en_l_DataCurrentState AS LOADING_HIGHER_NIPPLE_DATA_STATE
DECLARE u8_dataLoaded AS FALSE
DECLARE u8_l_DataSent_Bit AS FALSE
SWITCH en_l_DataCurrentState DO
CASE LOADING_HIGHER_NIPPLE_DATA_STATE:
u8_l_DataSent_Bit = FALSE
IF u8_dataLoaded IS NOT TRUE THEN
Set the upper 4 bits of DATA_REGISTER to (DATA_REGISTER AND 0x0F) OR (u8_data AND 0xF0)
Mark the data as loaded
END IF
Set RS_PIN of LCD_CMD_PORT to HIGH
Set EN_PIN of LCD_CMD_PORT to HIGH
IF delay_ms_UsingTimer(10) IS TIMER0_DELAY_DONE THEN
Set EN_PIN of LCD_CMD_PORT to LOW
Change the current state to LOADING_LOWER_NIPPLE_DATA_STATE
Reset the data loaded flag
END IF
CASE LOADING_LOWER_NIPPLE_DATA_STATE:
IF u8_dataLoaded IS NOT TRUE THEN
Set the lower 4 bits of DATA_REGISTER to (DATA_REGISTER AND 0x0F) OR (u8_data << 4)
Mark the data as loaded
END IF
Set EN_PIN of LCD_CMD_PORT to HIGH
IF delay_ms_UsingTimer(10) IS TIMER0_DELAY_DONE THEN
Set EN_PIN of LCD_CMD_PORT to LOW
Change the current state to LOADING_HIGHER_NIPPLE_DATA_STATE
Reset the data loaded and data sent flags
Set the data sent bit to TRUE
END IF
END SWITCH
RETURN u8_l_DataSent_Bit
END FUNCTION
```

- Flowchart :

