Bloqueos identificados y soluciones

1) En **smarthome.cpp**, hay un delay que es bloqueante, esto es porque el delay se usa para medir el tiempo que transcurre con la alarma activada en **strobelight.cpp.** También el delay es usado en **matrix_keypad.cpp** para el debounce de las teclas, donde timeincrement_ms es el tiempo de delay.

```
delay(SYSTEM_TIME_INCREMENT_MS);
```

```
void strobeLightUpdate( int strobeTime )
{
    static int accumulatedTimeAlarm = 0;
    accumulatedTimeAlarm = accumulatedTimeAlarm + SYSTEM_TIME_INCREMENT_MS;

    if( strobeLightState ) {
        if( accumulatedTimeAlarm >= strobeTime ) {
            accumulatedTimeAlarm = 0;
            strobeLight= !strobeLight;
        }
    } else {
        strobeLight = OFF;
    }
}
```

```
case MATRIX_KEYPAD_DEBOUNCE:
    if( accumulatedDebounceMatrixKeypadTime >=
        DEBOUNCE_KEY_TIME_MS ) {
        keyDetected = matrixKeypadScan();
        if( keyDetected == matrixKeypadLastKeyPressed ) {
            matrixKeypadState = MATRIX_KEYPAD_KEY_HOLD_PRESSED;
        } else {
            matrixKeypadState = MATRIX_KEYPAD_SCANNING;
        }
    }
    accumulatedDebounceMatrixKeypadTime =
        accumulatedDebounceMatrixKeypadTime + timeIncrement_ms;
    break;
```

Solución: Se propone usar dos timers, uno para el debounce y otro para el timealarm para solucionar este bloqueo.

Objeto utilizado de mbed: Timer

Métodos utilizados: .start(), .stop(), .reset() .read_ms()

Se eliminaron las variables que se utilizaban para medir el tiempo y simplemente se hizo uso de los métodos mencionados para medir el tiempo

2) Se identifica en el archivo **pc_serial_com.cpp** que la función pcSerialComStringRead() es considerablemente bloqueante porque espera recibir no uno sino varios caracteres de la pc. Hasta que no los reciba, el codigo no continua.

```
static void pcSerialComStringRead( char* str, int strLength )
{
    int strIndex;
    for ( strIndex = 0; strIndex < strLength; strIndex++) {
        uartUsb.read( &str[strIndex] , 1 );
        uartUsb.write( &str[strIndex] ,1 );
    }
    str[strLength]='\0';
}
```

Solución: Para evitar el bloqueo refactoriza la función commandSetDateAndTime en dos funciones, una que comienza el procesamiento del comando y otra que lo actualiza. Se agrega el comando al enum pcSerialComMode_t y en la función

pcSerialComUpdate se agrega el case para la actualización del procesamiento. Las funciones modificadas en la refactorizacion son :

* pcSerialComSetDateAndTimeUpdate()

* commandSetDateAndTime()

* pcSerialComUpdate()