

1. Project title:

Control and monitoring an industrial sensors using GSM protocol



Patachi Andreea-Iulia

patachiandreea@yahoo.com

Mertic Alexandru

mertic.alexandru@gmail.com

2. Abstract:

Using XMC4500 Relax Kit, the software we wrote is used in industrial application field in order to monitor certain sensors. It is specially designed for data acquisition and control in areas where there is no Internet access and it is easily suitable for remote controlled actuators.

3. Introduction, project aims and objectives:

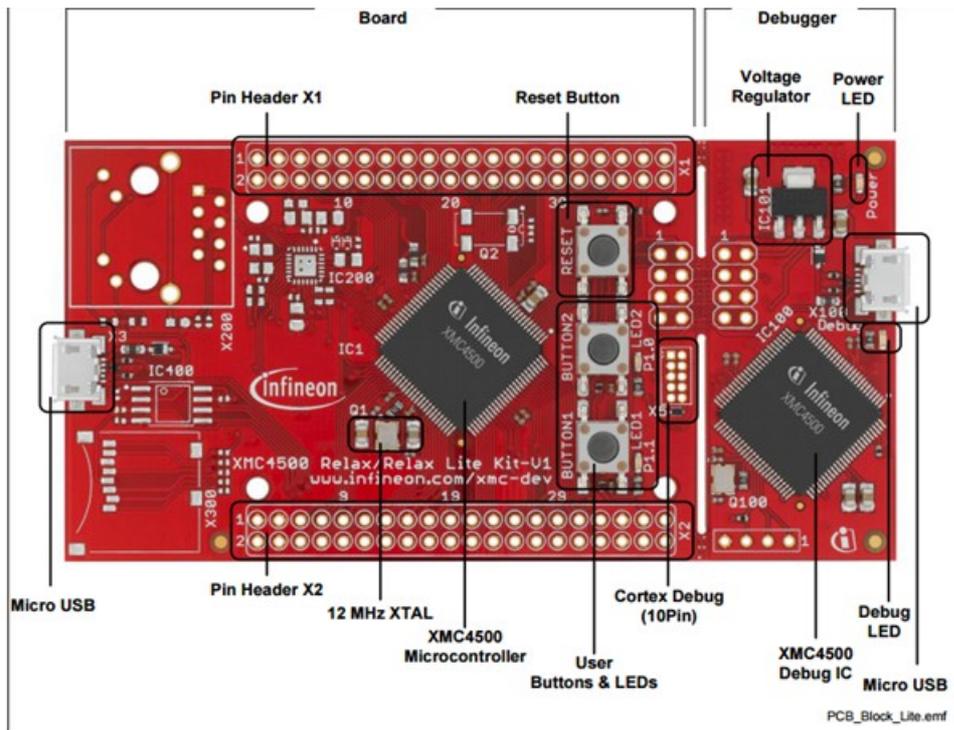
This project aims to provide a useful core device controller which can easily be extended in order to service more complex applications in industry and agriculture. There are industrial areas or forests, fields where agriculture is practiced without access to the Internet so SMS communication it is the best solution in such situations.

4. System overview:

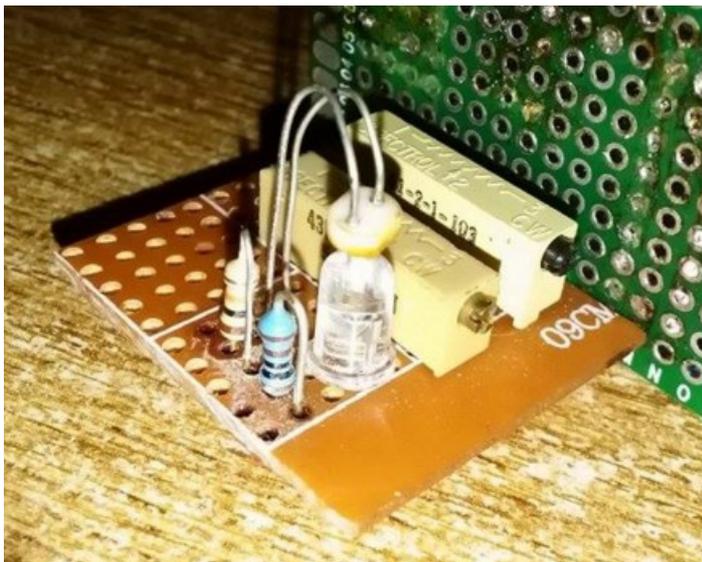
The system is completely made from the XMC4500 Relax Kit microcontroller (ARM® Cortex™-M4F based) connected with an an auxiliary board that acquires data from multiple sensors and also has a GSM module which is used for communication via SMS. Using DAVE 4.4.1 tool, we wrote code on the Relax Kit microcontroller to implement the desire tasks : monitoring , data acquisition and management of the information .

5. Schematics and components:

XMC4500 Relax Kit microcontroller



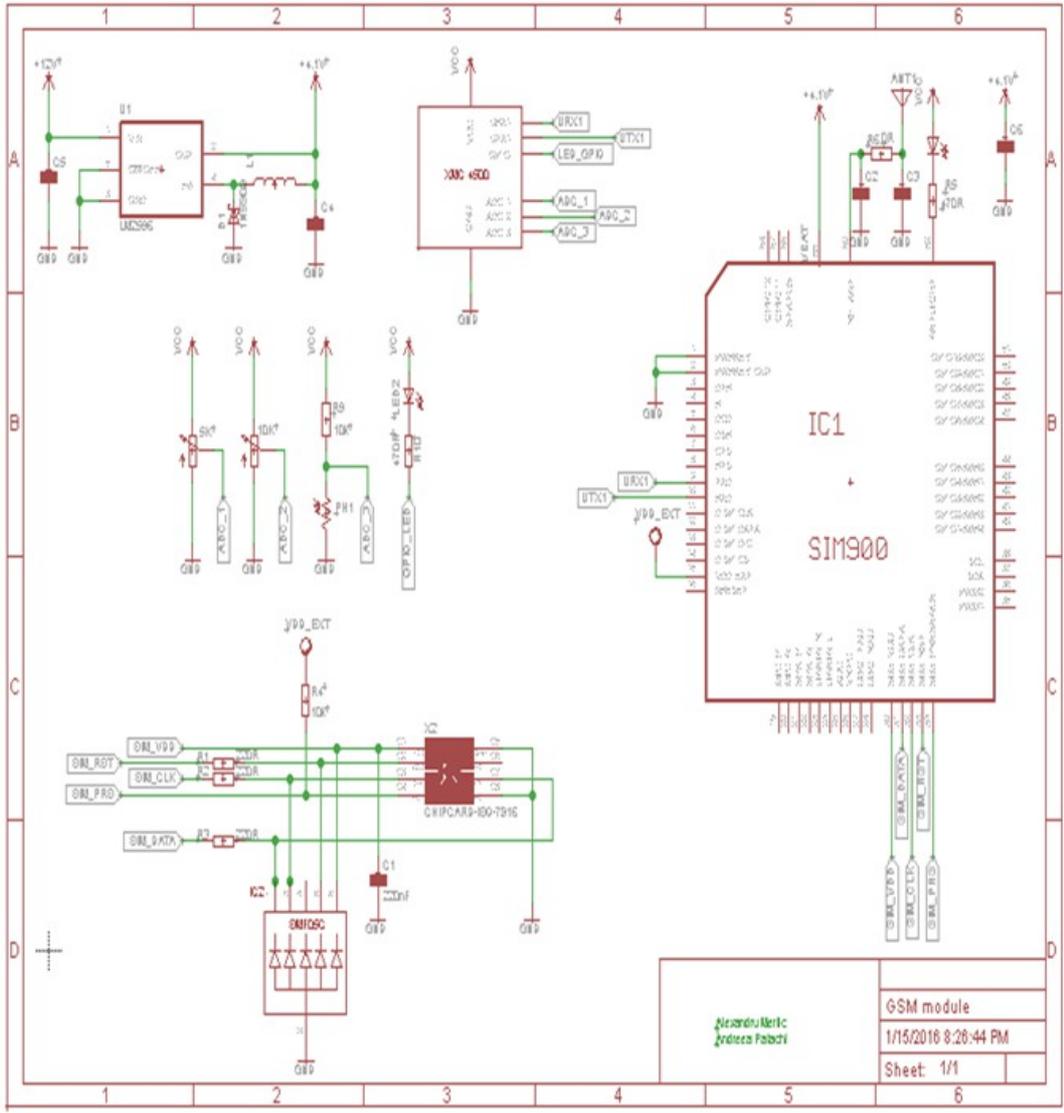
Acquisition board



GSM module

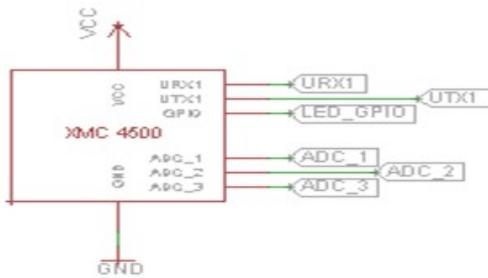


Board schematic



This contains :
 - GSM module

XMC4500 Relax Kit microcontroller



6. Software:

The software consists of C code, written using DAVE 4.4.1 IDE . In order to simulate some input-output sensors we have used two potentiometers, a fotorezistor and a LED in conjunction with ADC channels and GPIO respectively. During the development process we have used an USART channel to print debug useful information. The GSM functionality is highly important so we used the MVC pattern in order to keep the application very responsive, and also to avoid unwanted deadlocks. By adopting this pattern we easily integrated sensor data acquisition and control using just simple and accessible SMS.

Sensors data acuisiton

*/

```
typedef struct detailed_result_struct
```

```
{
```

```
    uint8_t channel_num;
```

```
    uint8_t group_num;
```

```
    uint16_t conversion_result;
```

```
} detailed_result_struct_t;
```

```
if((bool)(result >> VADC_GLOBRES_VF_Pos))
```

```
{
```

```
    valid_result = (bool>true;
```

```
    valid_result = (bool>true;
```

```
    detailed_result[index].channel_num = (result & VADC_GLOBRES_CHNR_Msk) >>  
    VADC_GLOBRES_CHNR_Pos;
```

```
    detailed_result[index].group_num = ADC_MEASUREMENT_Channel_C.group_index;
```

```
    //detailed_result[index].conversion_result = (result & VADC_GLOBRES_RESULT_Msk) >>
```

```
        // ((uint32_t)ADC_1.iclass_config_handle->conversion_mode_standard *
```

```
        (uint32_t)2);
```

```
    data_results.P1=(result & VADC_GLOBRES_RESULT_Msk) >>
```

```
        ((uint32_t)ADC_1.iclass_config_handle->conversion_mode_standard * (uint32_t)2);
```

```
}
```

```
    index++;
```

```
    result = ADC_MEASUREMENT_GetDetailedResult(&ADC_MEASUREMENT_Channel_B_handle);
```

```
if((bool)(result >> VADC_GLOBRES_VF_Pos))
```

```
{
```

```
    valid_result = (bool>true;
```

```
    valid_result = (bool>true;
```

```
    detailed_result[index].channel_num = (result & VADC_GLOBRES_CHNR_Msk) >>  
    VADC_GLOBRES_CHNR_Pos;
```

```

detailed_result[index].group_num = ADC_MEASUREMENT_Channel_B.group_index;
data_results.fotorezistor= (result & VADC_GLOBRES_RESULT_Msk) >>
    ((uint32_t)ADC_1.iclass_config_handle->conversion_mode_standard *
(uint32_t)2);
}
index++;
result = ADC_MEASUREMENT_GetDetailedResult(&ADC_MEASUREMENT_Channel_A_handle);
if((bool)(result >> VADC_GLOBRES_VF_Pos))
{
    valid_result = (bool>true;
    valid_result = (bool>true;
    detailed_result[index].channel_num = (result & VADC_GLOBRES_CHNR_Msk) >>
VADC_GLOBRES_CHNR_Pos;
    detailed_result[index].group_num = ADC_MEASUREMENT_Channel_A.group_index;
    data_results.P2= (result & VADC_GLOBRES_RESULT_Msk) >>
        ((uint32_t)ADC_1.iclass_config_handle-
>conversion_mode_standard * (uint32_t)2);
}
index++;
SMS analysis
static void parse_sms(void)
{
    sms_flow_ctrl.sms_rx = 0;
    if ( ( strstr((char *)received_sms, SMS_CMD_LED_OFF) != NULL ) ||
        ( strstr((char *)received_sms, SMS_CMD_LED_OFF_v2) != NULL ) ) {
        data_results.led = eLED_OFF;
    }
    if ( ( strstr((char *)received_sms, SMS_CMD_LED_ON) != NULL ) ||
        ( strstr((char *)received_sms, SMS_CMD_LED_ON_v2) != NULL ) ) {
        data_results.led = eLED_ON;
    }
    if ( ( strstr((char *)received_sms, SMS_CMD_STATUS) != NULL ) ||
        ( strstr((char *)received_sms, SMS_CMD_STATUS_v2) != NULL ) ) {
        sms_flow_ctrl.sms_tx = 1;
    }
    memset(received_sms, 0x00, sizeof(received_sms));
}
static void format_sms(char *payload)
{
    float p1, p2, p3;
    int p1_int, p1_fract, p2_int, p2_fract, p3_int, p3_fract;

```

```

p1 = GET_VOLTS(data_results.P1);
p2 = GET_VOLTS(data_results.P2);
p3 = GET_VOLTS(data_results.fotorezistor);
p1_int = (int)p1;
p1_fract = ((int)(p1 * 100)%100);
p2_int = (int)p2;
p2_fract = ((int)(p2 * 100)%100);
p3_int = (int)p3;
p3_fract = ((int)(p3 * 100)%100);
sprintf(payload, STATUS_SMS_PATTERN, p1_int, p1_fract, p2_int, p2_fract, p3_int, p3_fract,
(data_results.led == eLED_ON?"On":"Off"));
}

```

LED status refresh

```

void refresh_led_status(void) {
static LED_state_e last_led_status = eLED_OFF;
if ( data_results.led != last_led_status )
{
if ( data_results.led == eLED_ON )
{ DIGITAL_IO_SetOutputHigh(&LED); }
else { DIGITAL_IO_SetOutputLow(&LED); }
last_led_status = data_results.led;
}
}

```

7. Project results & applications :

This application is a useful idea and can be extended to a big project for industry , agriculture and many others things, even can be used at your own home in order to manage some important data only by sending an SMS.

8. References:

www.embedac.ro,

www.infineon.com

[www.simcom.us/act_admin/supportfile/SIM900_HD_V1.01\(091226\).pdf](http://www.simcom.us/act_admin/supportfile/SIM900_HD_V1.01(091226).pdf)

http://www.propox.com/download/docs/SIM900_AT.pdf