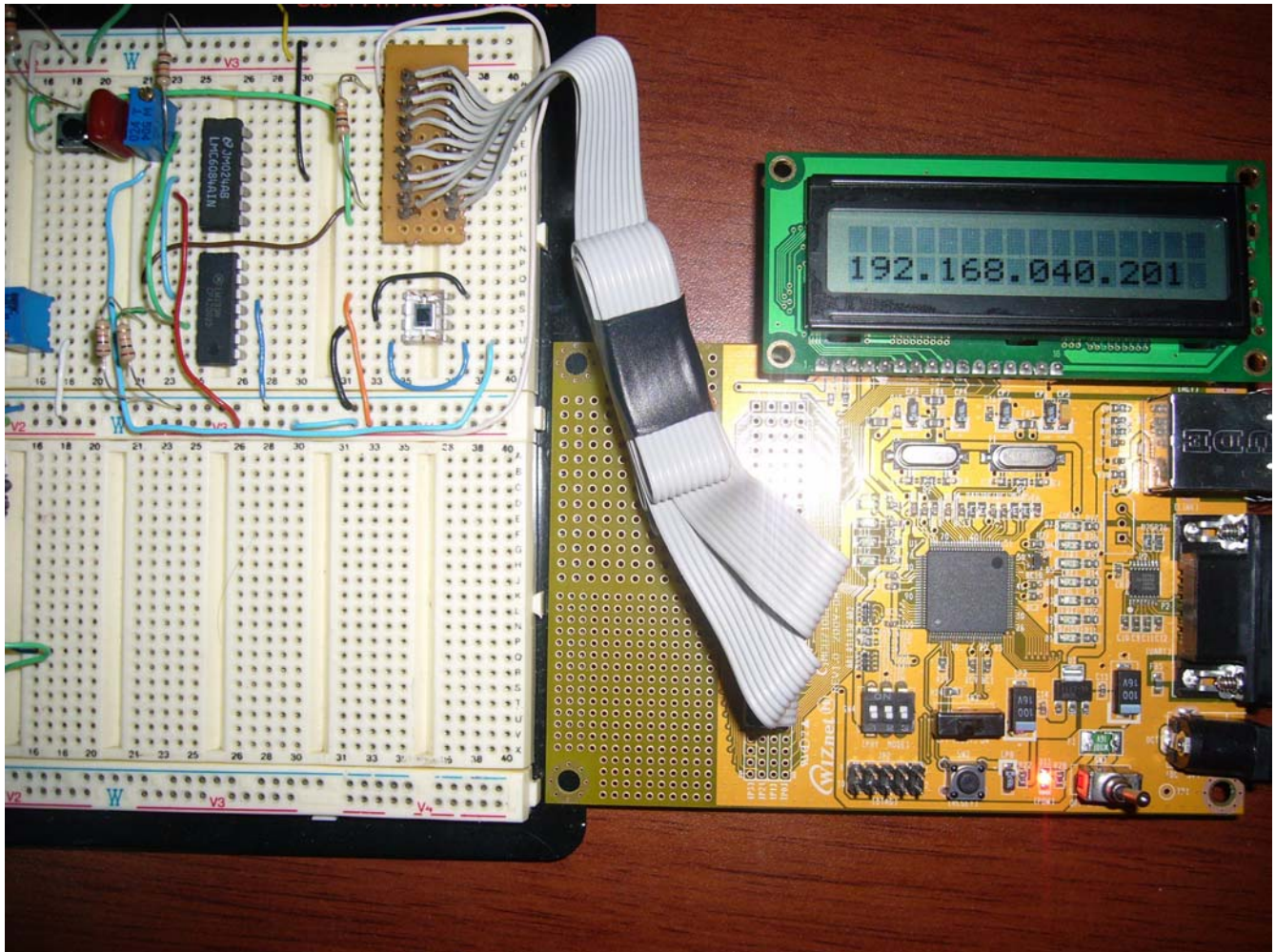
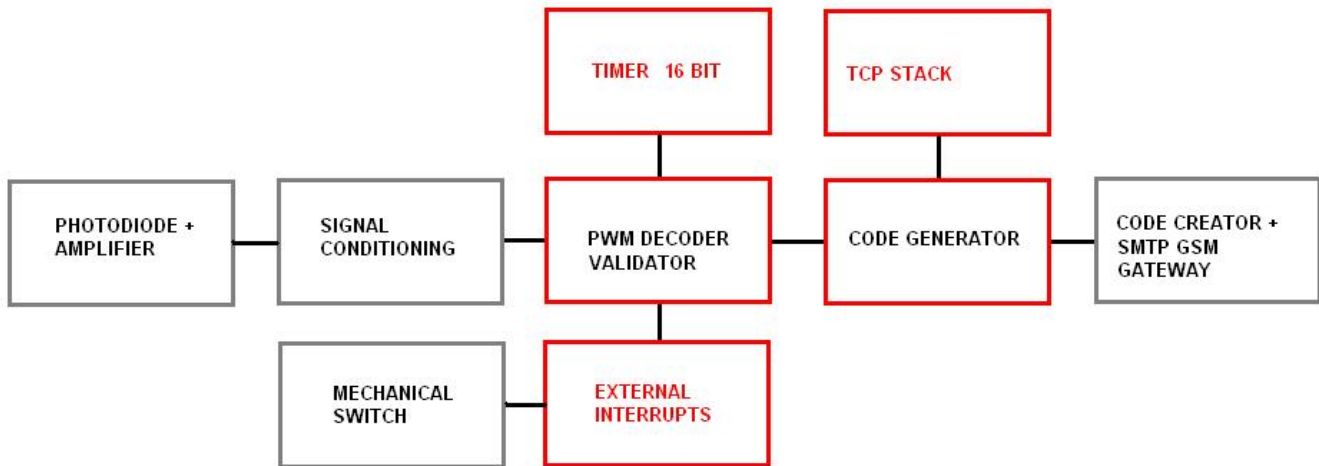


Photo Checkpoint

The Photo Checkpoint is an access control system that identify the user through a code represented by an animated picture showed on a cellphone screen so a W7100 with a light to voltage converter detect, decode and grant the access. The code is generated randomly by the W7100 and this communicate with a server to generate an image sequence. The picture code is transmitted to an e-mail account and a password to the cellphone through the GSM network.



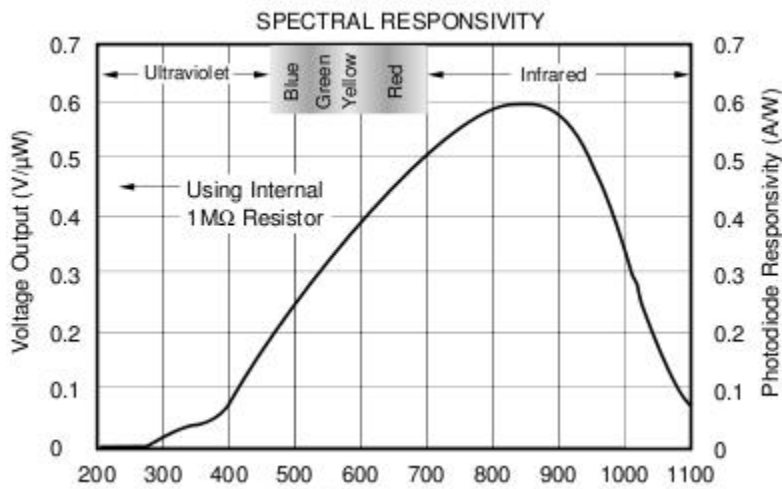
BLOCK DIAGRAM



The process start when a mechanical switch is pressed, this generates an access request. The 8051 processor from W7100 generates a code based on a random generator which is transmitted through a TCP socket to the Code Creator, in this case a PC running a script that generates the picture sequence. Once the animated picture is in the cellphone the file is opened and used as a key. The IC opt101 detect the code and the signal conditioning stage adapt the signal to be decoded by the pwm decoder and validator. The code is checked and the access is granted.

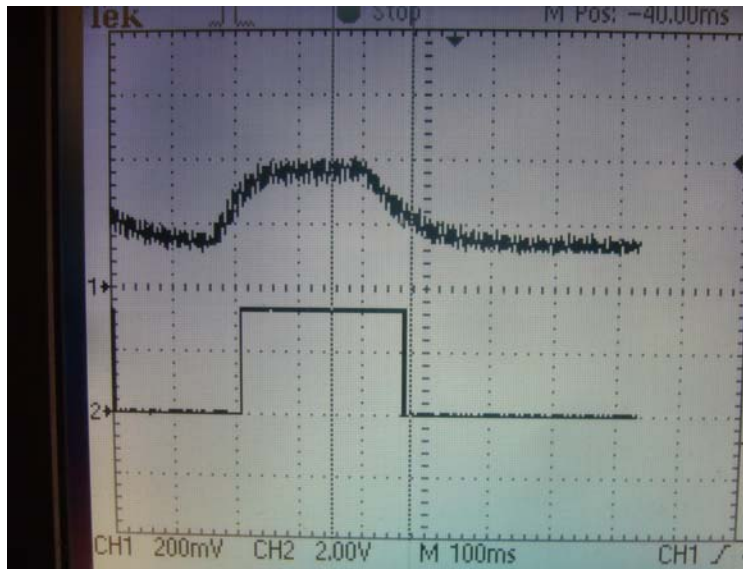
SIGNAL CONDITIONING

The IC OPT101 from TI is a light to voltage converter, it incorporates a photodiode and an amplifier. It is designed to get a linear response to light like next figure shows.



In the next picture, the signal on channel one of the oscilloscope come from the output of IC opt101.

The voltage is proportional to the light projected by the cellphone screen, in this case the signal correspond to a white picture. As you can see the signal requires amplification and filtering. I use to this purpose the operational amplifier LMC6084 which I used in previous projects with good results.



The signal showed in channel two is the signal after an amplifier, a low pass filter and finally a comparator implemented with an operational amplifier LM339.

EXAMPLE CODE

```
void pwmDecode (void)
{
    if((pwmCount > (PWM_START_BYTE - 1)) && (pwmCount < (PWM_START_BYTE + 1)) && pwmDecoding == 0) {
        pwmDecoding = 1;
        pwmCodeIdx = 0;
        console_print("\r\nDecoding...\r\n");
    }

    if(pwmDecoding == 1 && pwmNewByte == 1 && pwmCodeIdx < 9) {
        pwmCodeBuf[pwmCodeIdx++] = pwmCount;
        pwmNewByte = 0;
        sprintf (consoleBuf, "Nuevo Byte: %d\r\n", pwmCount);
        console_print(consoleBuf);
        if(pwmCodeIdx == 9) {
            pwmDecoding = 0;
            pwmNewCode = 1;
            console_print("\r\nNuevoCodigo...\r\n");
            cellpassCodeCheck();
            if (accessGranted == 8)
                console_print("\r\nAccess Granted!!");
        }
    }
}
```

