

Sviluppo di un sistema di controllo basato su microcontrollore per applicazioni fotovoltaiche soggette a partial shading

Progetto di Tesi di Laurea in Ingegneria Meccatronica

Referente: Prof. Paolo Magnone

Coreferente: Prof. Tommaso Caldognetto

Tesisti: Stefano Bozzolan, Paolo Franco, Gian Marco Pozzobon, Gabriele Scattolin

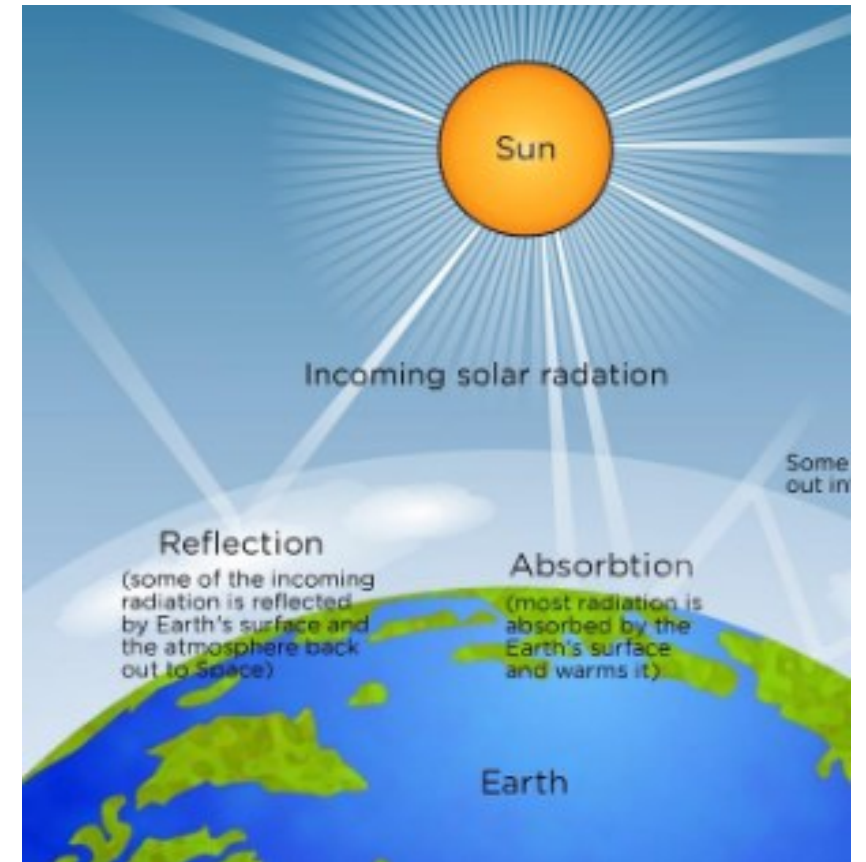
Obiettivi del progetto

- **Metodologia:** simulazione di diversi scenari di irraggiamento e temperatura.
- **Obiettivo principale:** sviluppare un sistema MPPT su microcontrollore per ottimizzare la produzione di energia anche in condizioni di ombreggiamento parziale.

Radiazione Solare

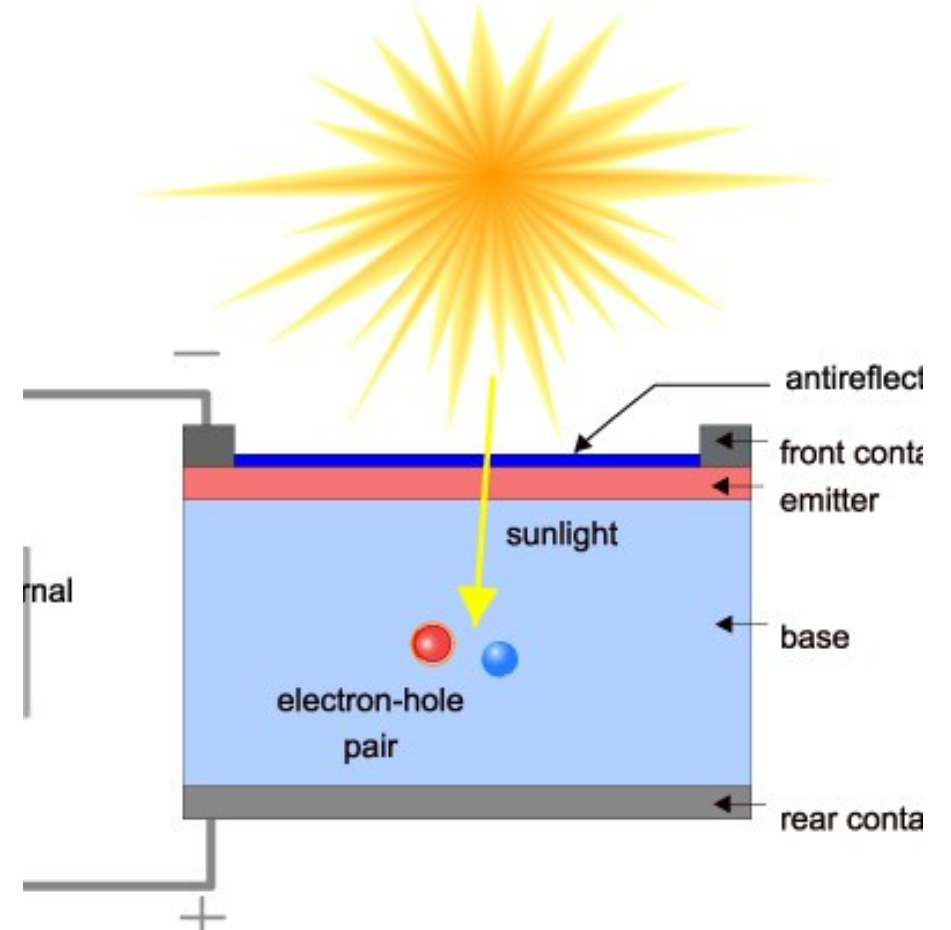
- **Punto principale:** l'irraggiamento è l'energia solare incidente sulla superficie terrestre, espressa in kWh/m²/giorno.
- **Costante solare:** Circa 1367 W/m².
- **Componenti della radiazione solare (G):** diretta (Gb), diffusa (Gd), riflessa (Ga)

$$G = G_b + G_d + G_a$$



Struttura delle celle solari

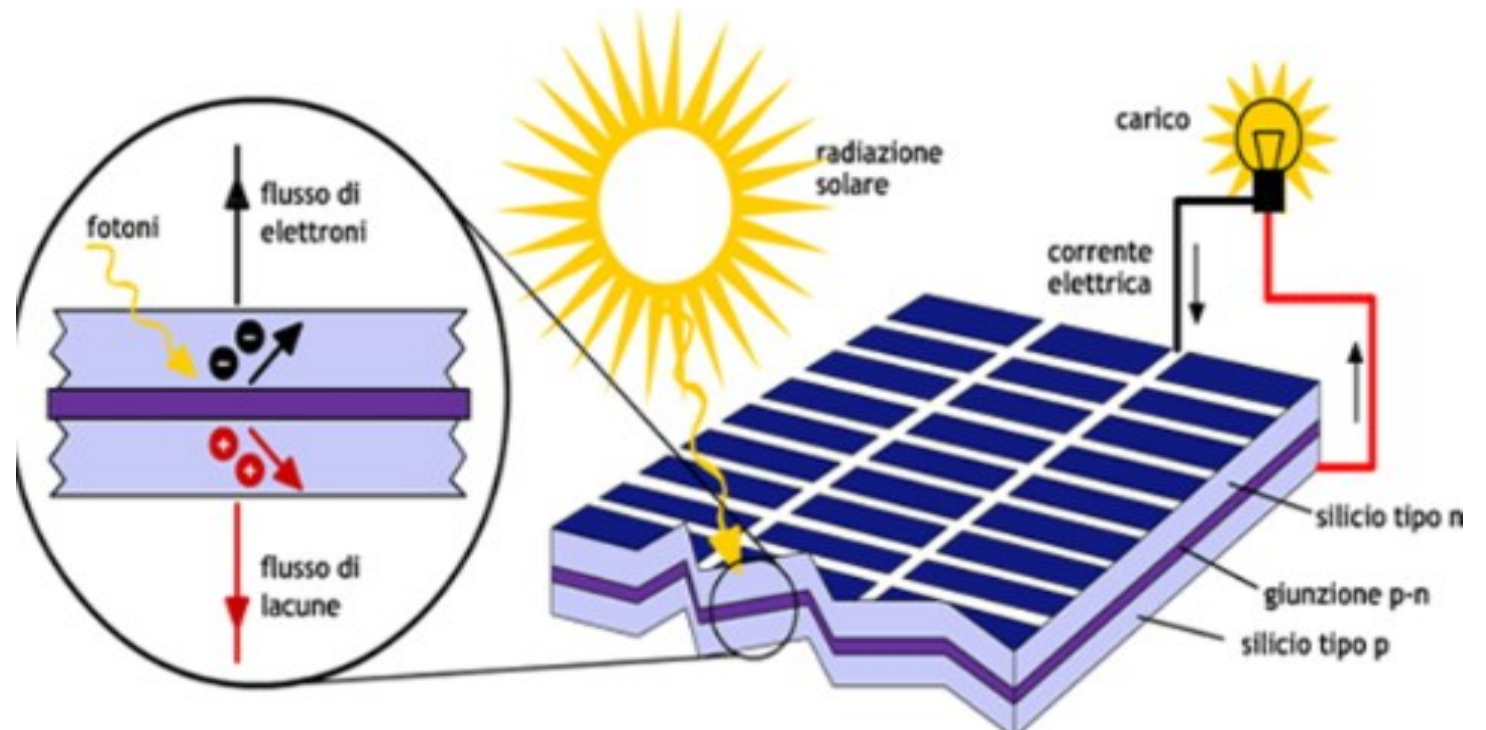
- **Dispositivi a giunzione p-n:**
separazione dei portatori
- **Assorbimento fotoni incidenti:**
generazione di coppia elettrone-lacuna.



Effetto Fotovoltaico

Effetto fotovoltaico:

l'effetto fotovoltaico è il fenomeno fisico per cui un materiale semiconduttore, come il silicio, genera una corrente elettrica quando viene colpito dalla luce solare. I fotoni della luce liberano elettroni nel semiconduttore, creando un flusso di carica.



Tipologie di Pannelli Fotovoltaici



Poly-Crystalline
Solar Cell



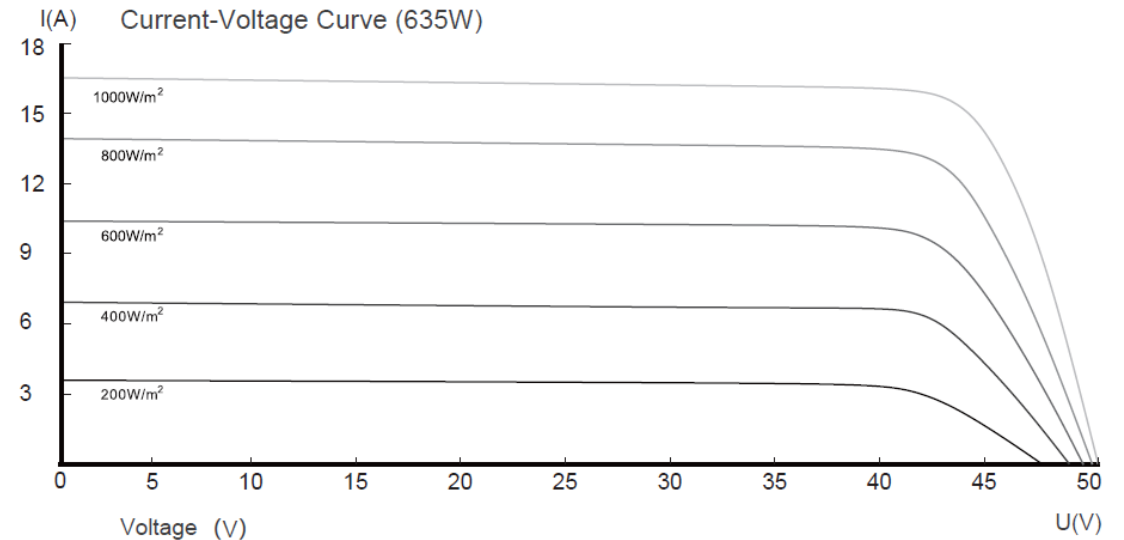
Mono-Crystalline
Solar Cell

- **Monocristallini:** alta efficienza, più costosi, colore scuro.
- **Policristallini:** meno efficienti, colore blu, più economici.
- **Amorfi (film sottile):** più economici, minor efficienza, ma versatili.

Modello Analitico del PV

- **Equazione della curva I-V del pannello fotovoltaico:**
parametri principali: V, I, T, G

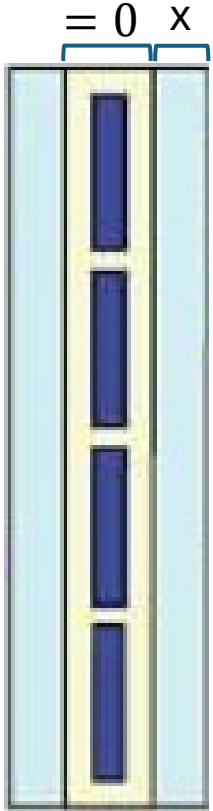
- $$V = \frac{n k T}{q} \ln\left(\frac{I_L - I}{I_0} + 1\right)$$
- $$P_{dissipata} = G A_{cella} (1 - n)$$
- $$I_L = I_{sc} + n_i(T_p - T_a) \frac{G}{G_{max}}$$
- $$T = P_{dissipata} H(s) + T_{ambiente}$$



Dipendenza dell'Irraggiamento e Modello Termico

- **Energia Solare:**
parte convertita in elettricità
parte dissipata come calore.
- **Effetto della Temperatura:**
aumento della temperatura -> riduzione dell'efficienza.
- **Dissipazione Termica:**
essenziale per mantenere le prestazioni ottimali.

Modello termico



$$T_{int} = P_{diss} * H(s) + T_a$$

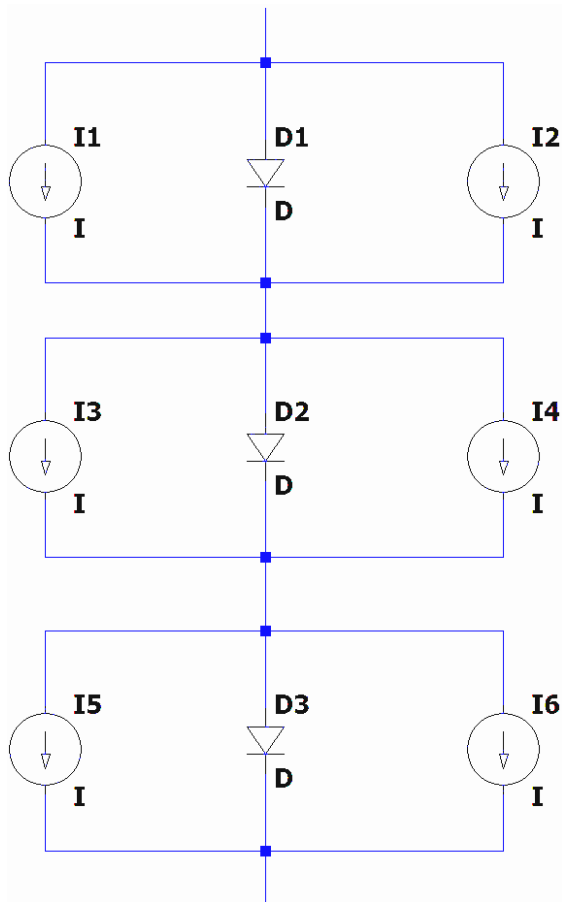
$$\left. \begin{aligned} q &= \frac{\lambda A \Delta T}{x} \\ q &= \alpha A (T_p - T_a) \end{aligned} \right\} T_{int} = \frac{q}{2} \left(\frac{x}{\lambda A} + \frac{1}{\alpha A} \right) + T_a$$

$$H(s) = \frac{1}{2} \left(\frac{x}{\lambda A} + \frac{1}{\alpha A} \right)$$

Connessione dei Moduli e Effetto del Partial Shading

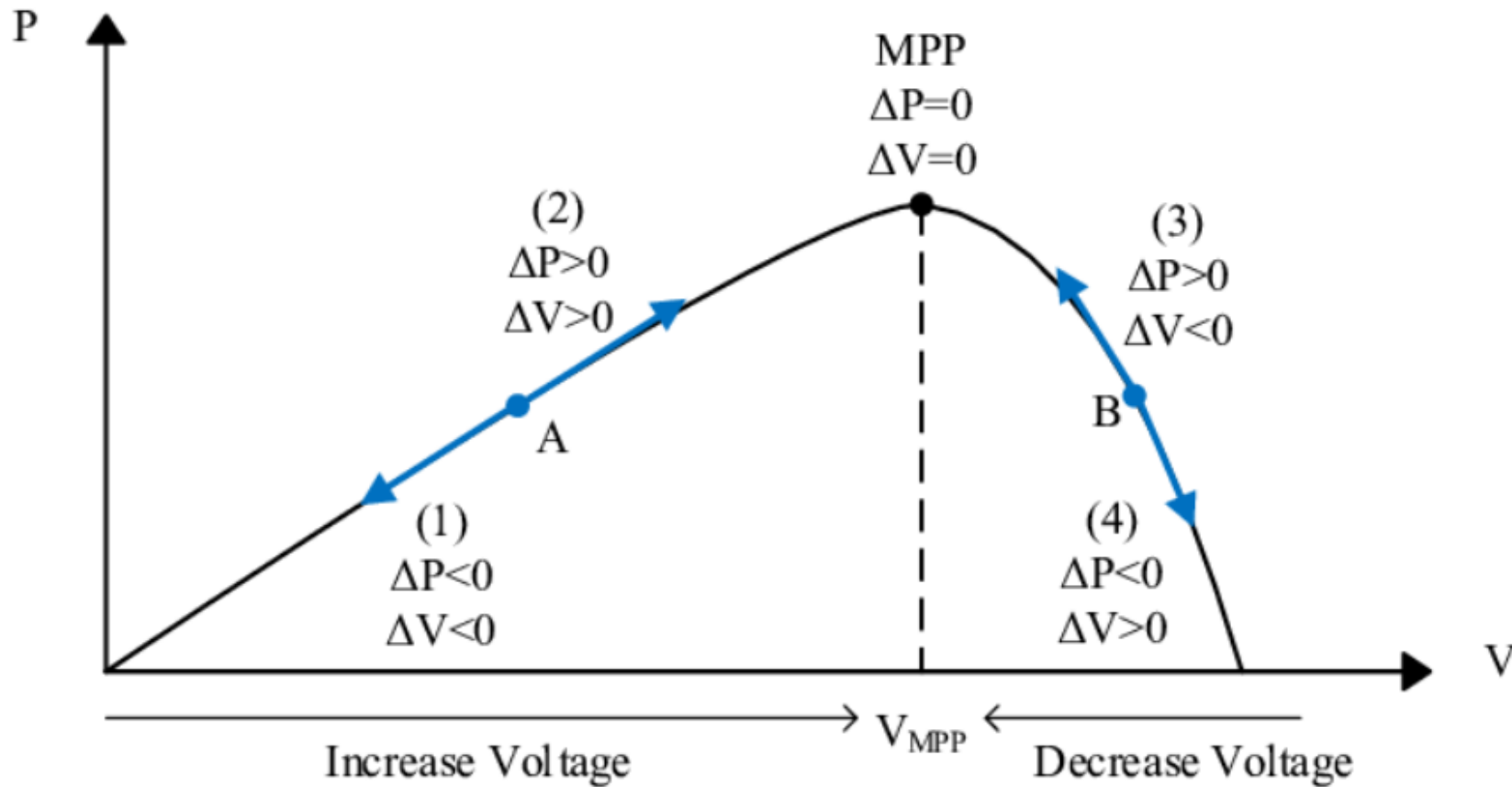
- **Problema del partial shading:** celle in ombra limitano l'output di tutto il modulo.
- **Ombreggiamento parziale:** quando una parte del pannello è in ombra, riduce la produzione complessiva di energia.
- **Funzione dei diodi di bypass:** i diodi di bypass permettono alla corrente di aggirare le celle ombreggiate, mantenendo una produzione di energia dalle celle non ombreggiate.
- **Effetto sui pannelli:** senza diodi, l'ombreggiamento può bloccare l'intero pannello. Con i diodi, solo le celle ombreggiate sono bypassate, migliorando l'efficienza complessiva.

Modulo PV utilizzato

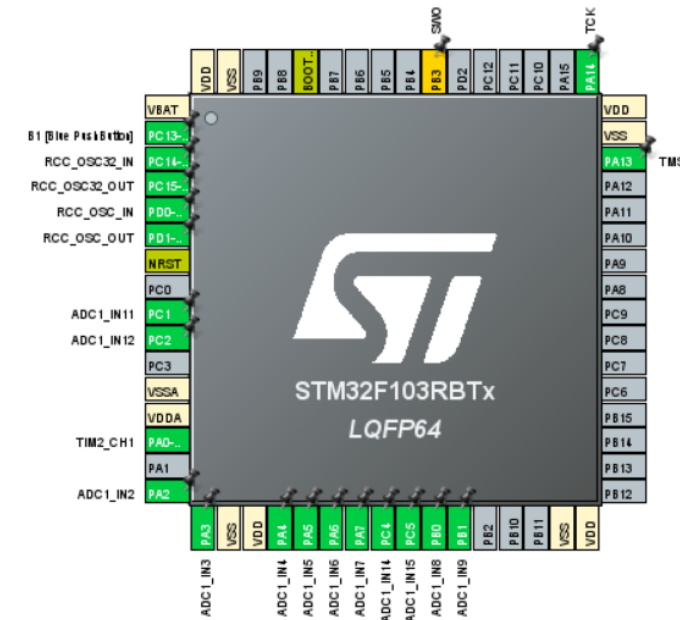


Module Type	DM610G12RT-B66HSW	
Testing Condition	STC ²	NMOT ³
Maximum Power (Pmax/W)	610	465
Maximum Power Current (Imp/A)	15.09	12.26
Maximum Power Voltage (Vmp/V)	40.45	37.95
Short-circuit Current (Isc/A)	15.99	12.89
Open-circuit Voltage (Voc/V)	48.69	46.86
Module Efficiency STC (%)	22.6%	

Algoritmo di Conduttanza Incrementale



Impostazione scheda



ADC1:

- 12 rank con 12 canali differenti
- Campionamento continuo
- Attivato dal trigger del Timer 3
- Utilizzo del DMA per trasferimento dati in memoria

Timer 2:

- Segnale PWM generato
- Prescaler: 5
- Counter Period: 65535

Timer 3:

- Prescaler: 60
- Counter Period: 65535
- Configurazione di Interrupt

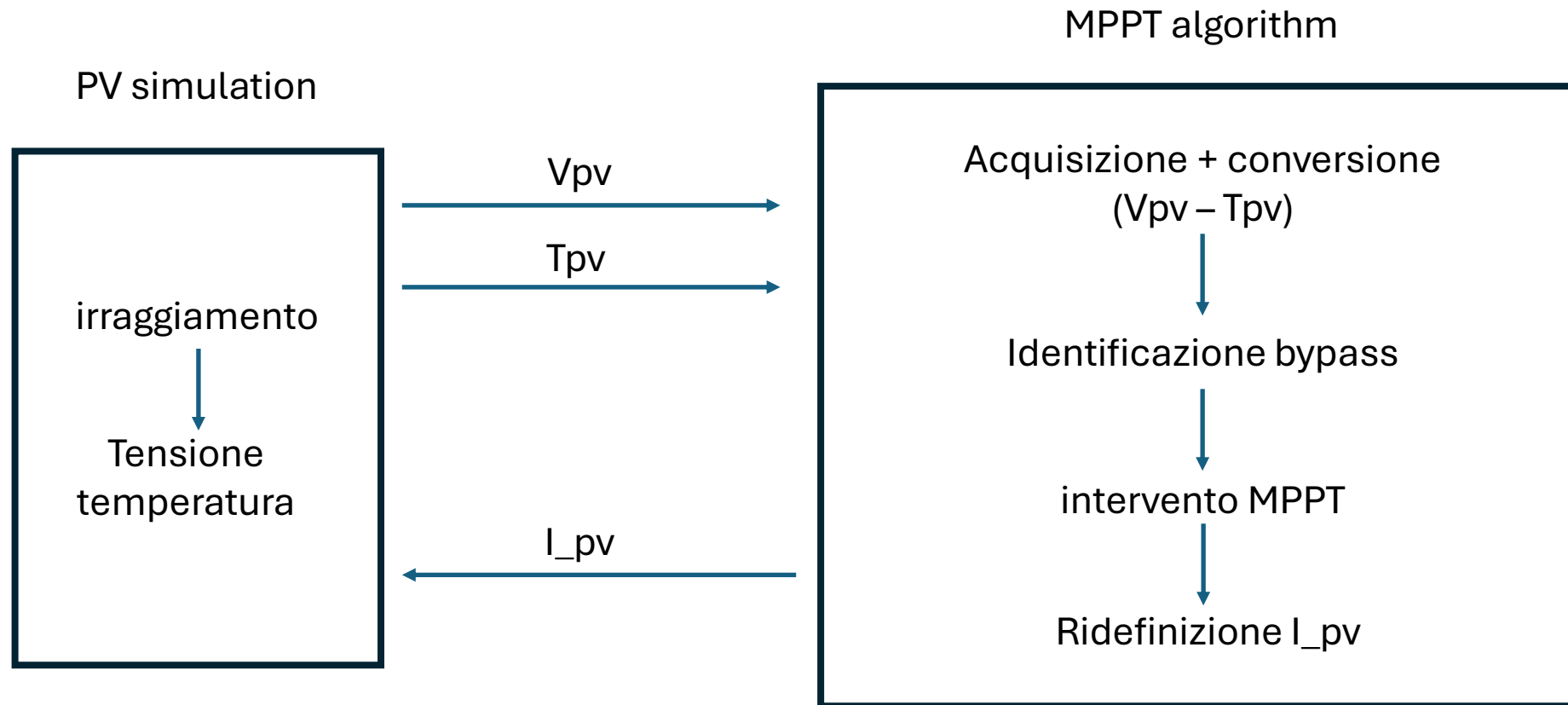
Frequenza di clock del sistema:

- 8 MHz → Interrupt ogni 0.5 secondi

Sviluppo dell'algoritmo MPPT

Codice e commenti

Struttura del sistema



Simulazione PV

```
while(1) {

    irraggiamento1 = 500 + 499 * __cos(0.5 * 3.1415926535 * frequenza * tempo);
    irraggiamento2 = 500 + 499 * __cos(1 * 3.1415926535 * frequenza * tempo);
    irraggiamento3 = 800.0;
    irraggiamento4 = 800.0;
    irraggiamento5 = 800.0;
    irraggiamento6 = 800.0;

    tempo = tempo + Tsample;

    I_pv=(PORTA&0x00000001);

    //modulo 1
    Pdissipata=irraggiamento1*AREA_CELLA*(1-EFFICIENZA)*22;
    Tpl = Pdissipata * 0.0753 + Tambiente;
    IL = (I_sc + EFFICIENZA_I*(Tpl-Tambiente))*irraggiamento1/1000;
    Vpl = (((n * k * Tpl / q) * (__log(((IL - I_pv) / I0)+1))*22) - 9) * k_sense;
    Tpl = (Tpl - 233)* k_temp;

    k_temp = 0.0264;// Coefficiente conversione temperatura per adc
    k_sense = 0.471;// Coefficiente conversione tensione per adc
```


MPPT – Acquisizione + conversione dati

```
void ADC1_2_IRQHandler(void)
{
    /* USER CODE BEGIN ADC1_2_IRQn 0 */

    HAL_ADC_Start_DMA(&hadcl, (uint32_t*)adc_buffer, 12);

    if (hadcl.Instance == ADC1)
    {
        // Copia dei dati dal buffer ai rispettivi array di tensione e temperatura
        for (int i = 0; i < N_CELLE; i++)
        {
            Vpv[i] = adc_buffer[2 * i]; // Dati di tensione
            Tpv[i] = adc_buffer[(2 * i) + 1]; // Dati di temperatura
        }

        // Conversione valori ADC in grandezze fisiche [V][K]
        for (int i = 0; i < N_CELLE; i++)
        {
            x_adc = Vpv[i];
            tens_conversion();
            Vpv[i] = x_adc;

            x_adc = Tpv[i];
            temp_conversion();
            Tpv[i] = x_adc;
        }
    }
}
```

MPPT – Acquisizione + conversione dati

```
#define propo_temp 0.03052503053 // [ (358 - 233) / 4095 ] Adattamento scala (FSR / #bit_adc)
#define propo_tens 0.001709402 // [ (16 - 9) / 4095 ] Adattamento scala (FSR / #bit_adc)

// Funzione conversione temperatura
void temp_conversion()
{
    x_adc = (x_adc * propo_temp) + 233;
}

// Funzione conversione tensione
void tens_conversion()
{
    x_adc = (x_adc * propo_tens) + 9;

    if ( x_adc < 10 )// Se minore di 10V (irraggiamento = 1), pongo a 0
    {
        x_adc = 0;
    }
}

// Esegui le funzioni di controllo necessarie

bypass_cells();
min_tens_research();
IL_value();
mppt_control();

// Aggiorna l'uscita PWM
pulse_value = I_pv;
```

Identificazione Bypass

```
void bypass_cells()
{
    Ttot = 0;
    for (int i = 0; i < N_CELLE; i++){// Temperatura media su tutte le celle
        Ttot += Tpv[i];
    }
    Tm = ((Ttot / N_CELLE) - 273)*0.9;// Converto in Celsius la media

    for (int i = 0; i < N_CELLE; i++) {
        Tpv[i] = Tpv[i] - 273;// Converto in Celsius il valore nell'array

        if(Tpv[i] <= Tm) {// Minore del 25% della temperatura media
            bypass ++; // Incremento la variabile 'bypass' tante volte quante le celle in shading
            Vpv[i] = 0; // Resetto valore array
            Tpv[i] = 0; // Resetto temperatura
        }
        else {// Tra le nuove correnti valide, scelgo la più piccola
            valid_cells_count++;
            Tpv[i] = Tpv[i] + 273;// Torno in Kelvin
        }
    }
}
```

Cella minor temperatura + Calcolo IL

```
void min_tens_research ()
{
    TempTest = Tpv[0];
    Vp_min = Vpv[0];
    Tp_min = Tpv[0];
    for(int i = 1; i < N_CELLE; i++){
        if( TempTest > 0 )
        {
            if ((Tpv[i] < TempTest) && (Tpv[i] > 0))
            {
                TempTest = Tpv[i];
                Vp_min = Vpv[i];
                Tp_min = Tpv[i];
            }
        }
        else
        {
            TempTest = Tpv[i];
            Vp_min = Vpv[i];
            Tp_min = Tpv[i];
        }
    }
}

void IL_value ()
{
    Pdiss_max = ((Tp_min - Tambiente ) / 0.0753 ) * 64; // Formato 10.6 (51 = 22098)
    irraggiamento = Pdiss_max / ((AREA_CELLA * (uno - EFFICIENZA ) * 22) >> 10); // Formato 16.0
    IL = ( ( ( I_sc + ( EFFICIENZA_I * ( Tp_min - Tambiente ) ) ) * irraggiamento ) << 3 ) / 1024 ;
}
```

Funzione MPPT

```
void mppt_control(){

    Vtot = 0;// Calcolo tensione totale del sistema
    for (int i = 0; i < N_CELLE; i++) { // Calcola la somma delle tensioni delle cell
        Vtot += Vpv[i];
    }
    Vtotale = Vtot;
    Power = Vtot * I_pv;// Calcola la potenza totale del sistema
    delta_P = Power - P_prev;
    delta_V = Vtot - V_prev; //(_sqrt((Vtot - V_prev)*(Vtot - V_prev)));

    dP_dV = (delta_V != 0) ? (delta_P / delta_V) : 0; // Calcolo del rapporto dP/dV

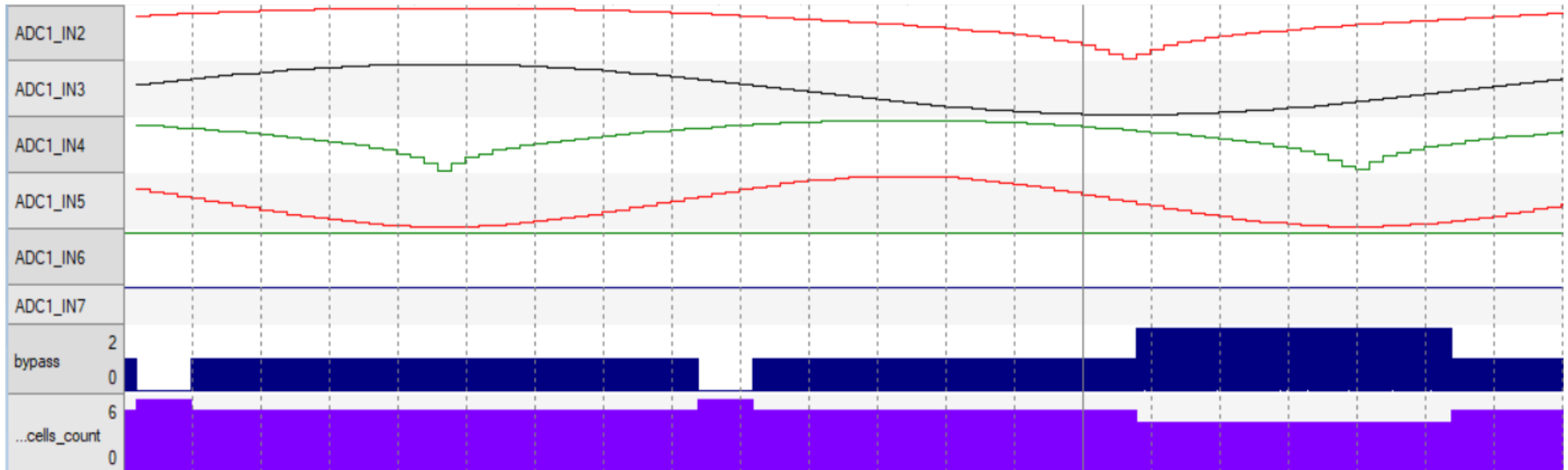
    // Aggiornamento della tensione basato su dP/dV e celle in bypass
    if (dP_dV > 0) // Pendenza positiva (ci troviamo a sx del punto di max P.) quindi
    { //Piccoli Passi
        Vtot += adjustment_step;// Fattore di correzione normale, per assenza di cel
    }
    else// Pendenza negativa (ci troviamo a dx del punto di max) quindi devo decreme
    {
        if (dP_dV < 0) // Pendenza negativa (ci troviamo a dx del punto di max P.) qui
        { //Piccoli Passi
            Vtot -= adjustment_step;// Fattore di correzione normale, per assenza di c
        }
        // Nel caso in cui la dP_dV = 0, non devo correggere niente (siamo già nel pun
    }
}
```

Funzione MPPT

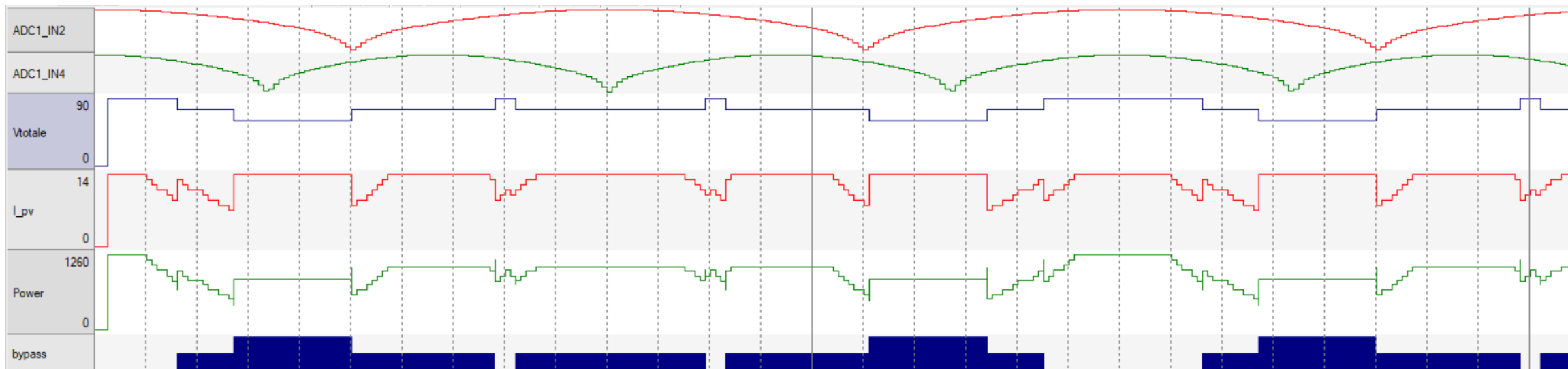
```
// Ricalcolo Ipv
Ttot = 0;
for (int i = 0; i < N_CELLE; i++)// Temperatura media celle valide
{
    Ttot += Tpv[i];
}
Tm = Ttot / valid_cells_count;

a = exp(((Vtot)/(n_k_q * Tm))>>1);// Formato .31
I_pv = (IL - (((IO>>1) * (a - s))>>20))>>11;// Formato 16.0

// Aggiornamento delle variabili per il prossimo ciclo
V_prev = Vtot;
P_prev = Power;
bypass = 0;
valid_cells_count=0;
Vprecedente = V_prev;
}
```



Ingressi pin, celle valide, celle in bypass



Tensioni, tensione totale, corrente imposta, potenza, celle in bypass

Conclusioni

- Difficoltà incontrate
 - Sviluppo dell'algoritmo di simulazione: Il linguaggio del codice non era chiaro inizialmente.
 - Associazione delle uscite del file di simulazione con gli ingressi corretti del file STM.
 - Recupero della corrente IL nel file STM: Non potendo inviarla tramite ADC.
 - Implementazione del bypass: Definire i criteri corretti per riconoscere il bypass.
- Risultati
 - Monitoraggio continuo delle variabili chiave.
 - Regolazione dinamica del sistema fotovoltaico.
 - Migliorata efficienza energetica complessiva dell'impianto.