EV Battery Charging Control Using DSP Controller

Motilal Nehru National Institute of Technology, Allahabad

End Semester Evaluation

Oct, 2024

Under the supervision of Prof. Rajesh Gupta
MNNIT Allahabad

Group No.19
Pratham Raj Bhatt (20212048)
Ashish Kumar Verma (20212031)
Pratiksha Anuragi (20212093)

CONTENTS

Introduction

- Project Goal
- Key Technology
- Software Development
- Safety Focus
- Advanced Features

Problem Formulation

The problem formulation involves addressing several key challenges:

- Need for Efficient Charging
- Precise Control
- Safety Concerns
- Adaptability
- Complexity

Motivation

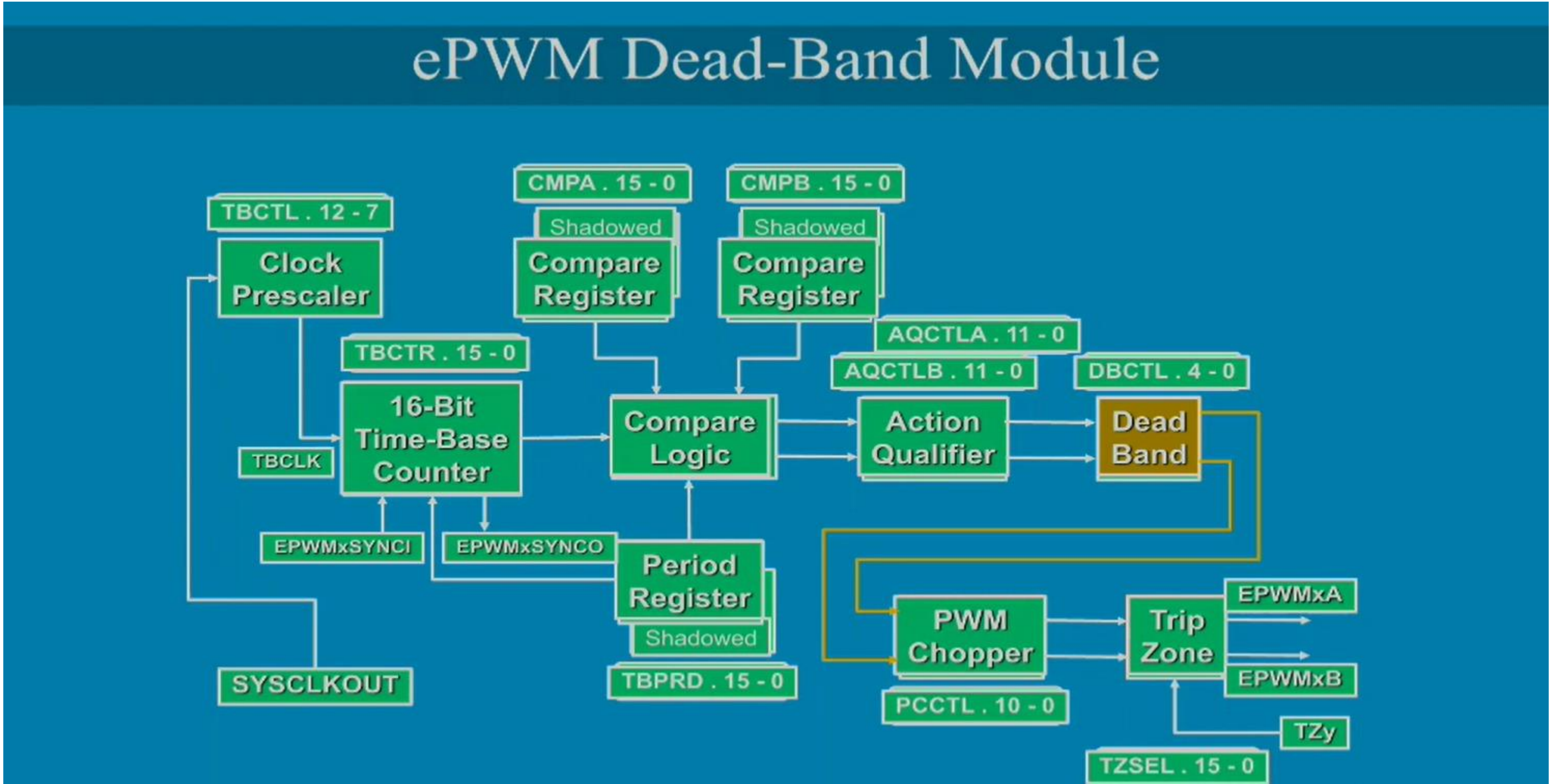
- Growing EV Market
- Need for Efficiency
- Safety Concerns
- Technological Advancement

Objectives

The primary objective is to develop an efficient and sustainable EV battery charging system to enhance energy conversion:

- Develop a Functional Charger
- Master ePWM Module
- Utilize Code Composer Studio
- Prioritize Safety

METHODOLOGY




= compare events are fed to the Action Qualifier Module
TBCTR
TBPRD
CMPA
CMPB
Asymmetrical Waveform
Count Up Mode
TBCTR
TBPRD
CMPA
CMPB
Asymmetrical Waveform
Count Down Mode
TBCTR
TBPRD
CMPA
CMPB
Symmetrical Waveform
Count Up and Down Mode

```
interrupt_void cpu_timer0_isr(void){


	static int up_down =1;
	CpuTimer0.InterruptCount++;
	EALLOW;
	SysCtrlRegs.WDKEY = 0xAA;
	EDIS;
	if(up_down){

		if(EPwm1Regs.CMPA.half.CMPA<EPwm1Regs.TBPRD){
			EPwm1Regs.CMPA.half.CMPA++;
		}
		else{
			up_down =0;
		}

	}
	else{

		if(EPwm1Regs.CMPA.half.CMPA>0){
			EPwm1Regs.CMPA.half.CMPA--;
		}
		else{
			up_down =1;
		}

		PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
	}

}
```

```
void applyPhaseShift(float* input, float* output, int bufferSize, float phaseShiftRadians) {
    float delaySamples = (phaseShiftRadians / (2 * PI)) * SAMPLE_RATE;
    int integerDelay = (int)delaySamples;
    float fractionalDelay = delaySamples - integerDelay;

    for (int i = 0; i < bufferSize; i++) {
        int delayedIndex = (bufferIndex - integerDelay + 1024) % 1024;
        float delayedSample = input[delayedIndex];

       //LI
        int nextIndex = (delayedIndex + 1) % 1024;
        delayedSample += fractionalDelay * (input[nextIndex] - input[delayedIndex]);

        output[i] = delayedSample;
        bufferIndex = (bufferIndex + 1) % 1024;
    }
}
```

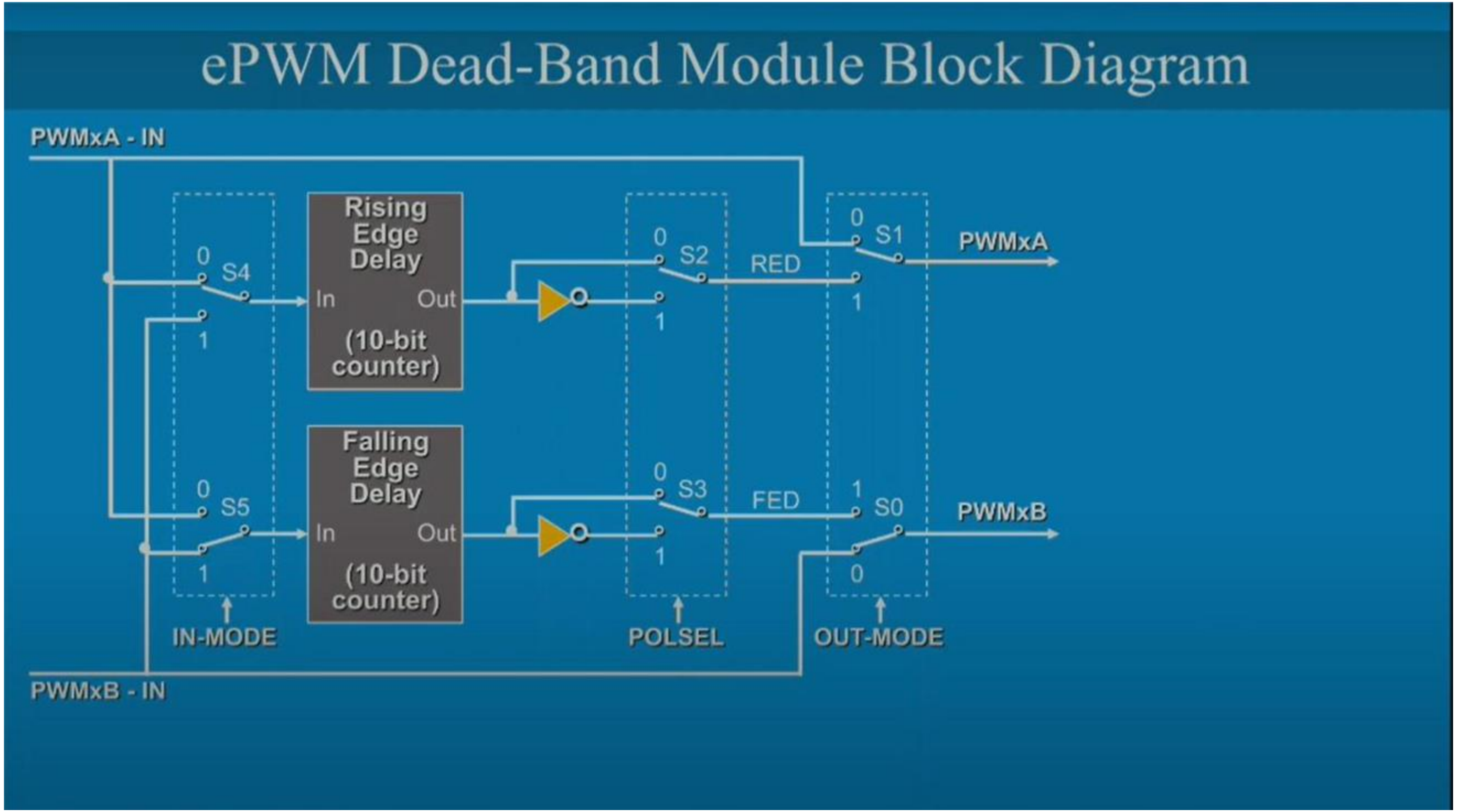
```c
void UpdatePWMDutyCycle(float dutyCycle) {
    EALLOW;


    if (dutyCycle > 100.0) dutyCycle = 100.0;
    if (dutyCycle < 0.0) dutyCycle = 0.0;


    EPwm1Regs.CMPA.bit.CMPA = (uint16_t)((dutyCycle / 100.0) * EPwm1Regs.TBPRD);

    EDIS;
}
```

```
void InitPWM() {
    EALLOW;


    EPwm1Regs.TBPRD = TB_CLK / PWM_FREQUENCY;
    EPwm1Regs.TBPHS.bit.TBPHS = 0;
    EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN;
    EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;  /
    EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;
    EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;


    EPwm1Regs.CMPA.bit.CMPA = EPwm1Regs.TBPRD / 2;


    EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
    EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;

    EDIS;
}
```

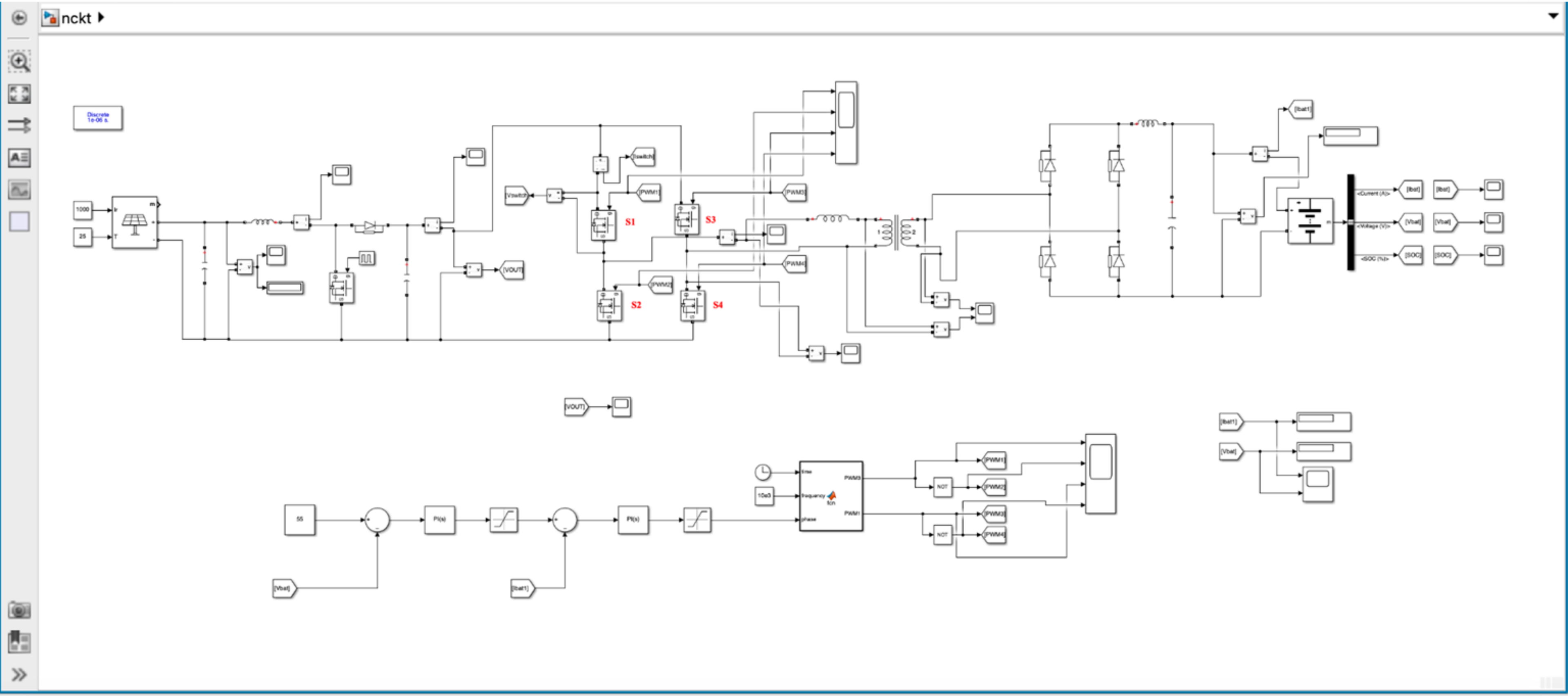
Name	Description	Structure
DBCTL	Dead-Band Control	EPwm*x*Regs.DBCTL.all =
DBRED	10-bit Rising Edge Delay	EPwm*x*Regs.DBRED =
DBFED	10-bit Falling Edge Delay	EPwm*x*Regs.DBFED =

Rising Edge Delay = T_{TBCLK} x DBRED

Falling Edge Delay = T_{TBCLK} x DBFED

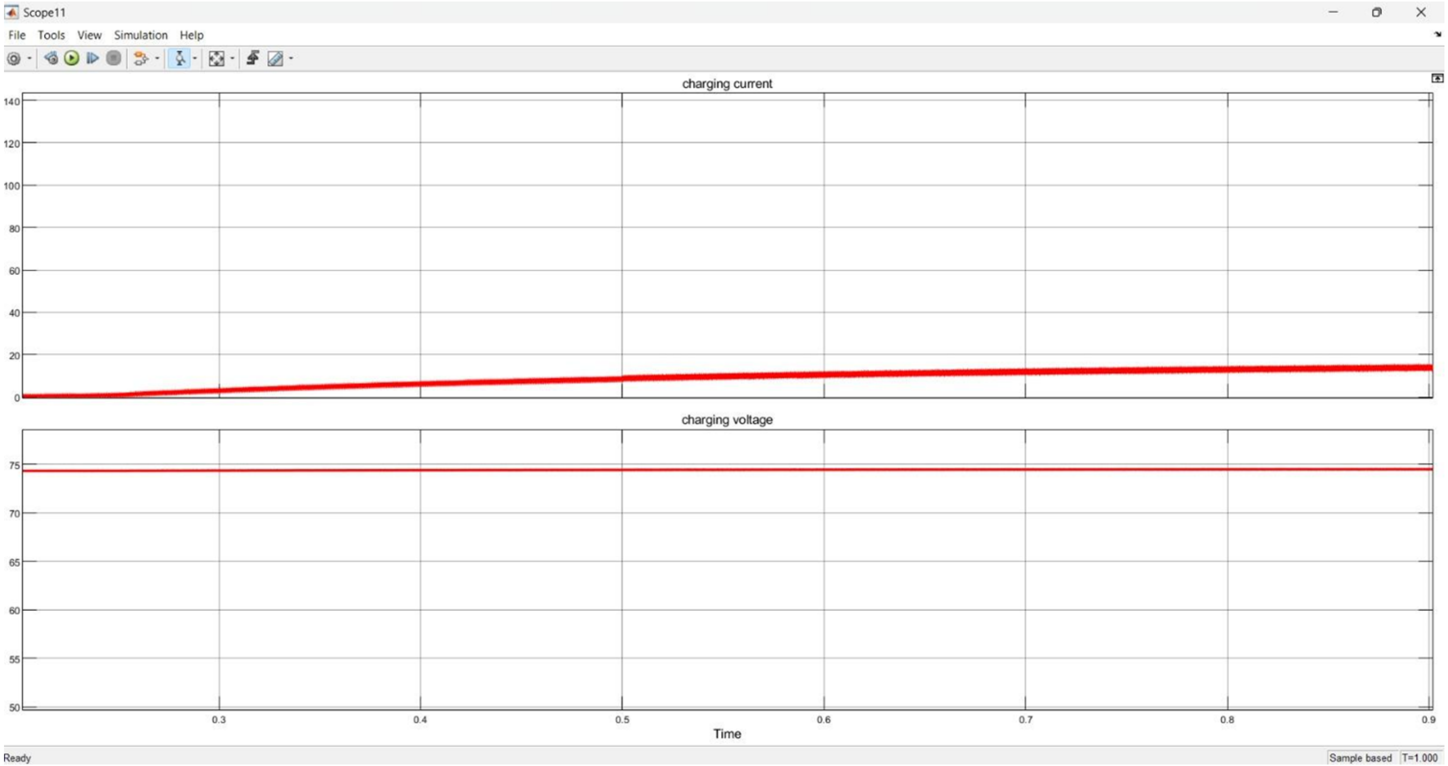

ePWM Dead-Band Module Block Diagram
PWMxA - IN
0
S4
1
Rising Edge Delay
In
Out
(10-bit counter)
0
S2
1
RED
0
S1
1
PWMxA
0
S5
1
Falling Edge Delay
In
Out
(10-bit counter)
0
S3
1
FED
1
S0
0
PWMxB
IN-MODE
POLSEL
OUT-MODE
PWMxB - IN

Simulation Model

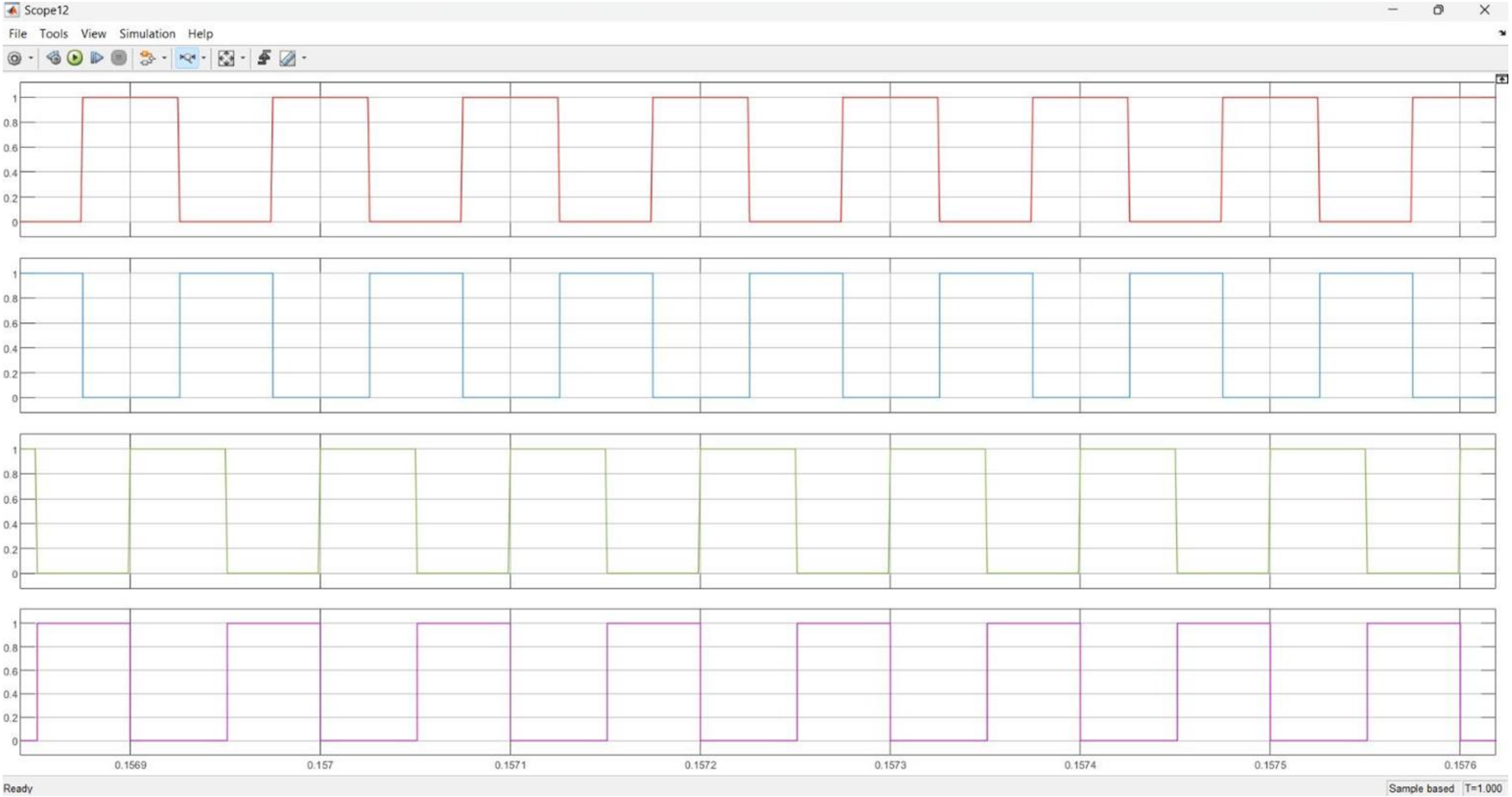


```
function [PWM3,PWM1] = fcn(time,frequency,phase)
Tswitching = 1/frequency;
PWM1 =0;
PWM3 =0;
y1 = mod(time,Tswitching);
if y1 < Tswitching/2
    PWM1 = 1;
end
t_phi = Tswitching*phase/360;
y2 = mod(time+t_phi, Tswitching);
if y2 < Tswitching/2
    PWM3 = 1;
end
```

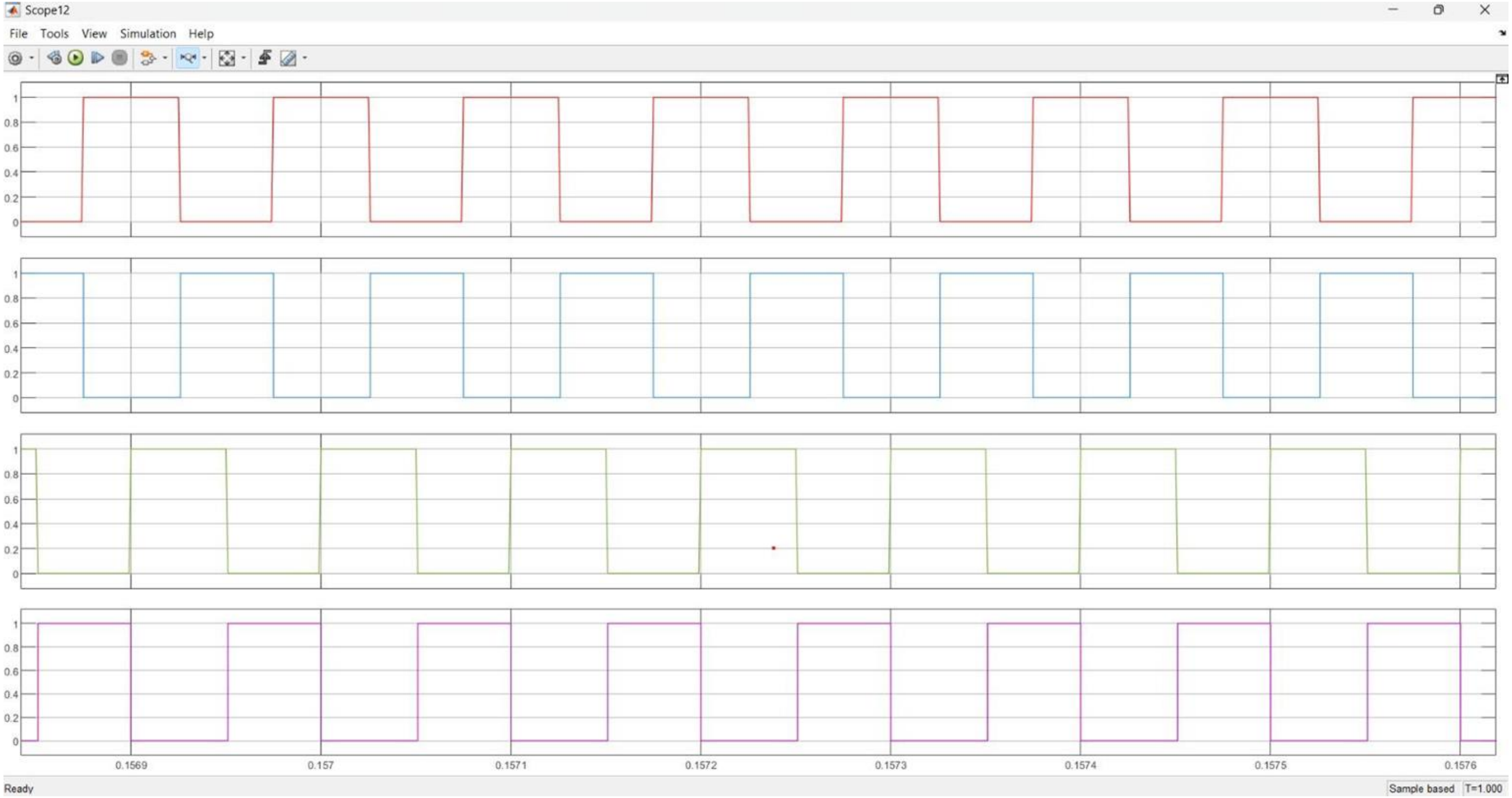
Results



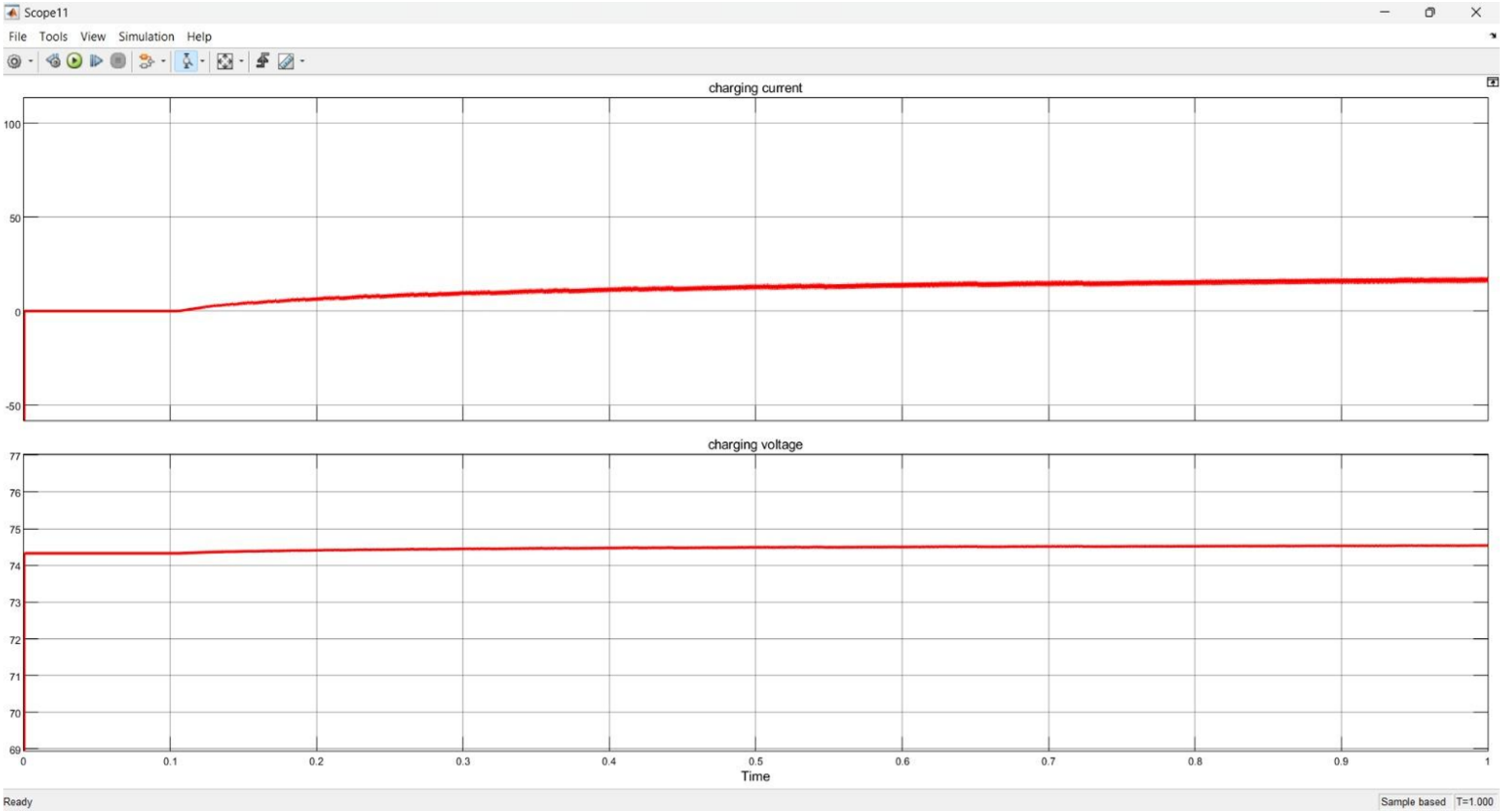
i) at SOC 10%



ii) phase 90 degree at Kp= Ki = 1



iii) phase 150 degree at Kp=5 and Ki=1



iv.) At Kp=5 and Ki = 1

CONCLUSION

This project successfully designed and implemented an EV battery charger using a DSP controller and Code Composer Studio. By harnessing the advanced features of the ePWM module, we achieved precise control over the charging process, ensuring both efficiency and safety.

Key Takeaways

- ePWM Expertise
- Algorithm Implementation
- Safety Focus
- Advanced Techniques

REFERENCES

1Soumya Ranjan Meheí and Rajeev Kumaí Singh, "A Standaíd ľwo Stage On-Boaíd Chaígeí With Single Contíolled PWM and Minimum Switch Count" IEEE ľRANSACľIONS ON INDUSľRY APPLICAľIONS, VOL. 59, NO. 4, JULY/AUGUSľ 2023

2D. Boweímasteí, M. Alexandeí and M. Duvall, “ľhe need foí chaíging: evaluating utility infíastíuctuíes foí electíic vehicles while píoviding customeí suppoít,” IEEE Electíification Magazine, vol. 5, no. 1, pp. 59- 67, Maích 2017.

3A. K. Singh., K. A. Chinmaya, and M. Badoni, “Solaí PV and gíid based isolated conveíteí foí plug-in electíic vehicles”. IEľ Poweí Electíonics, no. 12, pp. 3707-3715, 2019.

4A. K. Singh, A. K. Mishra, K. K. Gupta, P. Bhatnagar and T. Kim, "An integrated converter with reduced components for electric vehicles utilizing solar and grid power sources," IEEE Trans. on Transp. Electrification, vol. 6, no. 2, pp. 439-452, June 2020.

WORK PLAN

- Literature survey
- Data collection of necessary specifications, IEEE standards and
  parameters
- Modelling and Simulation
- Data analysis
- Optimization

THANK YOU