MICRO- PHASOR MEASUREMENT UNIT (μPMU) DEVELOPMENT AND IT’S APPLICATIONS

Report submitted to
Indian Institute of Technology, Kharagpur
for the award of the degree

of

Bachelor of Technology
in Electrical Engineering

by

Hari Prasanna Das

Under the guidance of

Prof. A.K. Pradhan

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR
May 2016

CERTIFICATE

This is to certify that the thesis entitled "**Micro- Phasor Measurement Unit Development and its Applications**" submitted by **Hari Prasanna Das (12EE10017)** for the award of the degree of **Bachelor of Technology (Hon's)** is a record of bona fide work carried out by him under our supervision and guidance. This thesis has fulfilled the requirements according to the regulations of this institute and in our opinion, has reached the necessary standard for submission. The work done in this thesis has not been submitted to any other University or Institute for award of any degree or diploma.

Prof. Ashok Kumar Pradhan
Department of Electrical Engineering
Indian Institute of Technology Kharagpur
Kharagpur, 721302, India.
Date:

Declaration

I certify that

a. the work contained in this report is original and has been done by me under the guidance of my supervisor(s).

b. the work has not been submitted to any other Institute for any degree or diploma.

c. I have followed the guidelines provided by the Institute in preparing the report.

d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Hari Prasanna Das (12EE10017)
Department of Electrical Engineering
Indian Institute of Technology Kharagpur
Kharagpur, 721302, India.
Date:

Acknowledgement

I, Hari Prasanna Das, 4th Year Undergraduate student of the Department of Electrical Engineering, Indian Institute of Technology (IIT), Kharagpur had immense pleasure working for this project.

At the outset, I would like to thank Prof. A. Routray, B.Tech Project Coordinator, whose guidance was invaluable during the coursework of the project.

I would like to give my heartfelt gratitude to Prof. A.K. Pradhan, whose kind help and proper guidance paved the way for successful completion of the project. I will remain ever indebted for his endeavours towards me.

I specially thank all the PhD Scholars of Power Systems, Department of Electrical Engineering, IIT Kharagpur for their immense help.

Lastly, I would like to thank all the associated staffs of Department of Electrical Engineering who were directly or indirectly involved in the successful accomplishment of the project. The support was without a parallel.

Hari Prasanna Das (12EE10017)
Department of Electrical Engineering
Indian Institute of Technology Kharagpur
Kharagpur, 721302, India.
Date:

Abstract

With mostly radial power distribution and one-way power flow, it was only necessary to evaluate the envelope of design conditions, e.g., peak loads or fault currents, rather than continually observe the operating state. But the growth of distributed energy resources introduces variability, uncertainty, and opportunities to recruit diverse resources for grid services. Multiple resources on each feeder cause more complex impacts on the circuit behavior that can be observed with voltage and current phase angle variations.

To address the resulting need for tools to better observe, understand and manage the grid at the distribution scale, there is a need to develop a high-precision measurement system. In this field, Phasor Measurement Units (PMUs) are available for transmission systems with time synchronized measurement. In current scenario, Phasor Measurement Units (PMUs) are positioned mainly on transmission system or in sub-stations. The novelty of PMUs can also be best utilized in the case of distribution systems. But keeping the difference of transmission and distribution systems in mind, a special type of PMU, called μPMU will be required in latter case.

This paper investigates the properties of a μPMU taking distribution system constraints and the possible sources of measurement errors into account. Feasible solutions to the distribution system problems are provided. It also presents a prototype of a μPMU using Microcontrollers for Phasor estimation and Wifi Development Board for time signal acquisition and synchronization. At last, possible applications of μPMU data is analyzed.

Index Terms: Phasor Measurement Units, Synchrophasor, Micro-PMU, Power Distribution, Smart Grid

Contents

Chapter 1

Introduction

Phasor Measurement Unit that provides time synchronized phasor data is now a mature technology and is capable of realizing many applications related to power systems.[1] PMUs are devices which sample voltages and currents through voltage and current transformers. The devices then estimate the voltage and current phasors in reference to the timing signal from global positioning system (GPS). The phasors are time-stamped using the GPS time and transmitted through communication network. Based on their accurately measured data and synchronized network, many PMUs deployed in a wide area are connected together and able to estimate the dynamic status of the whole power system. This wide area measurement system is capable of power system stability monitoring, post-mortem analysis, adaptive protection and control, etc. Currently, PMUs are utilized in state estimation, linear optimal control, oscillation control, adaptive protection, event location, dynamic line rating, etc.

When it comes to application of synchrophasor technology to power distribution system, a specialized PMU (called μPMU) is required. This is done keeping in view of the differences in measurement parameters in transmission and distribution network. In distribution system the voltage angle difference beaten two adjacent buses may be few millidegrees compared to degree in case of transmission system. This paper first analyses the problems that are needed to be catered in case of a distribution system. It also considers the possible sources of measurement errors and their theoretical and practical limits. Based on above constraints, solutions are formulated to overcome them. A prototype of μPMU is also presented and is being developed as a part of research work. It is being kept in mind as to make the μPMU less and less costly keeping the model in par with protocols. Some optimization is also done on and over the model proposed by [2][3] so as to achieve a sound business case for the μPMUs.

1.1 Literature Survey

PMU is an electronic device which provides time-tagged positive sequence voltage and current phasor, frequency, rate of change of frequency with the help of Global Positioning System Satelite (GPS). GPS facilitates time synchronization of measured signal at geographically dispersed locations. This helps to obtain a comprehensive view of power system at an instant when the measurements were taken. Block diagram of a general PMU is shown in Fig. 1.1. Analog Input consists of 3-Phase voltages and currents available from Potential Transformers (PTs) and Current Transformers (CTs). In order to filter out the frequencies above the Nyquist rate, from the input signals, anti-aliasing filter is used. Filtered signals are sent to analog to digital converter (ADC) to obtain digital samples. The ADC works in phase-lock with GPS pulses. The phase-lock oscillator converts the GPS signal at 1 pulse per second into required high-speed timing pulses used in the waveform sampling. The microprocessor computes positive sequence voltage and current using techniques reported in [4]. Finally the positive sequence components are time stamped and uploaded to Phasor Data Concentrator (PDC) through communication interface according to IEEE data transmission standard [5].

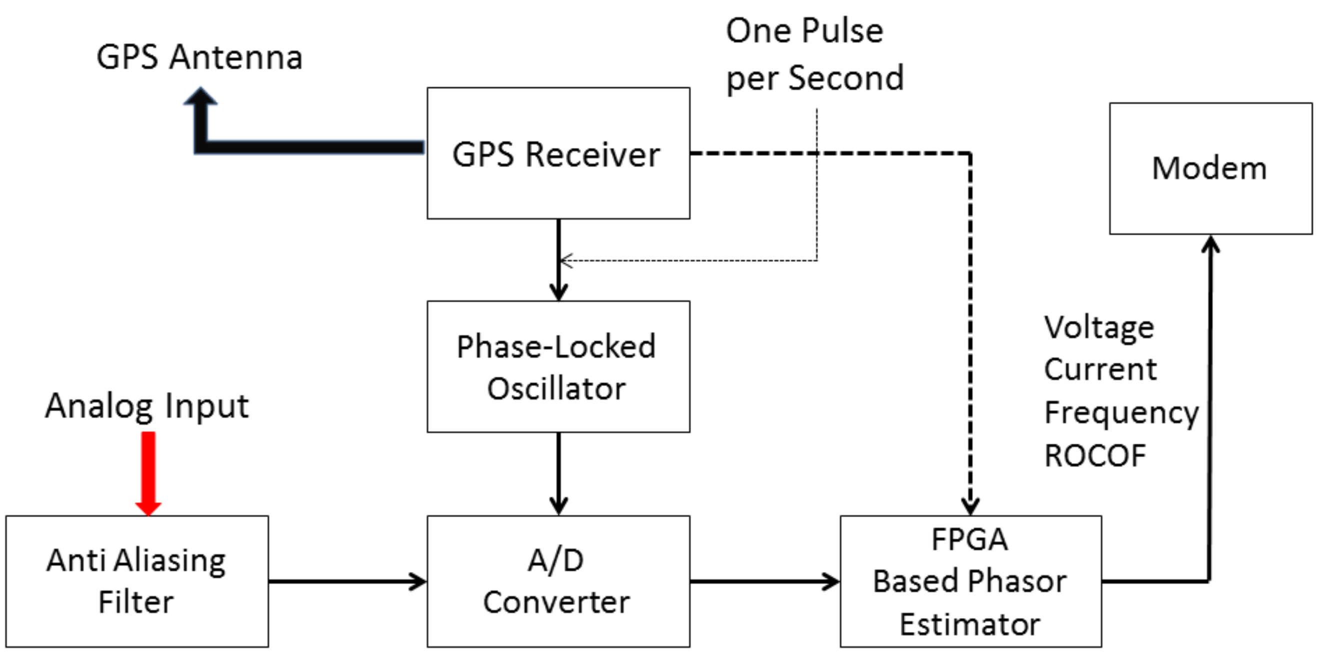


Figure 1.1 Block Diagram of a generic Phasor Measurement Unit

Need of specialized PMU for Distribution System:

Today, synchrophasors are almost used exclusively to observe transmission system. To apply the synchrophasor technology to distribution system, the problems of distribution system must be analyzed first. Distribution application of synchrophasor technology is more challenging because of the following reasons:

- Voltage angle differences between locations on a distribution circuit will be up to two orders of magnitude smaller than those on transmission network
- Various instrumental errors and high white noise affect the distribution system measurements to a greater extent unlike transmission system where the percentage error is less.
- The costs of PMUs must be lower in order to have a sound business case for installation of multiple PMUs in distribution circuit as compared to transmission network.

Hence, there is a need for a specialized PMU for distribution system applications.

Sources of Measurement Error

The sources of measurement error can come from Instrumental constraints, noise or harmonics. All measurement errors can be listed as (Fig. 1.2):

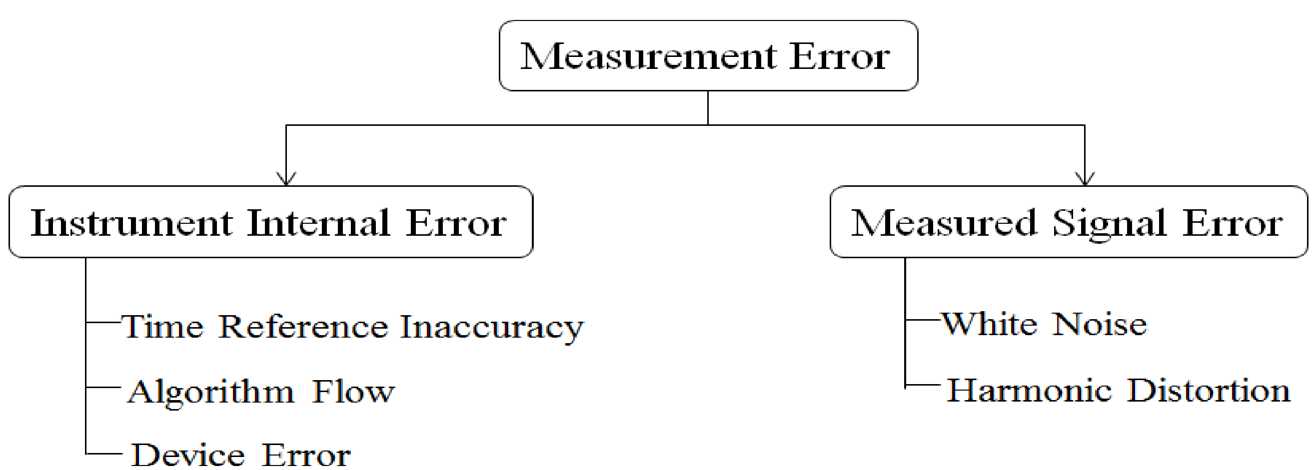


Figure 1.2. Measurement Errors Classification

Global Positioning System (GPS) is usually used to provide the time reference for synchrophasors, taking advantage of its accuracy and synchronicity [6] The pulse per second (PPS) signal, which is supplied by most GPS receivers, provides a high precision 1 s periodical signal whose rising edge is aligned to the Coordinated Universal Timing (UTC) second boundary. This feature enables the wide-area measurement of PMUs to obtain a precise and synchronized frequency reference. However, this PPS signal is not ideal in reality, due to the transmission delay between GPS satellites and receiver, and the interferences and noise on the receiver.

Two measures of the signal should be considered: accuracy and precision. Accuracy means how close the signal rising edge is to the actual GPS second boundary. It can often be a few microseconds off from the actual GPS timing, but this error is usually fixed for a given firmware thus can be calibrated. Precision is how much the PPS signal edge changes from one second to the other. This change is also called the PPS jitter. The GPS satellite geometry, signal conditions, drift and variation of the oscillator, as well as the internal circuit of the GPS receiver all contribute to the jitter [7]. Due to the uncertainty of trace delay, interference and oscillator precision, the GPS jitter varies and cannot be eliminated by calibration. A synchrophasor measurement algorithm is used in the instrument to estimate frequency and angle of power grid signal.

Approximations are often utilized by the algorithm for processing simplification and easier realisability, thus bringing in error.

Device error refers to the error caused by the instrument hardware, including internal noise, analog-to-digital (ADC) error, rounding error, etc. Here the ADC error is considered as the main error source. When an analog signal is sampled and converted to digital signal by ADC, the continuous values of voltage are rounded to their nearest digital value due to the resolution and accuracy of the ADC, thus error is brought. Furthermore, the offset and nonlinear error of ADC are also added into the total ADC error.

The intensity of noise is evaluated by signal-to noise ratio (SNR), which is defined as the ratio of the signal power to the noise power and given in the form of logarithm by the following formula

$$SNR_{dB} = 10\ log_{10} \frac{P_{signal}}{P_{noise}} \qquad (1)$$

Harmonic distortion is one of the major noise in the power system, and it affects the accuracy of frequency and phase angle measurement. The total harmonic distortion (THD) factor is the ratio of the RMS values of all of the harmonic components together to the RMS amplitude of the fundamental component as follows [8]:

$$THD(\%) = \frac{\sqrt{\sum_{n=2}^{\infty} i_{sn}^2}}{i_{s1}} \times 100\% \quad (2)$$

1.2 Motivation

With the fast growth of distributed energy sources, Wide Area Monitoring System (WAMS) for distribution system has a great scope. The base component of WAMS is an accurate and reliable measurement unit. In this sphere, Synchrophasor Technology proves the best. The time synchronization allows for better and better forms of monitoring over a large system. In current scenario, Phasor Measurement Units are mostly used in transmission system. They fail to capture the intricacies of distribution system unless they are customized for specific use. So, much scope is open to understand the distribution system measurement requirements and thus design a measurement unit to cater the need for synchronized measurements in distribution system. This project deals with such a specialized PMU called Micro-PMU (µPMU) and its applicability for distribution system measurements.

1.3 Objective

The objective of the project is to

- Understand the distribution system problems that rule out the possibility of using the same measurement unit for distribution and transmission systems
- Formulate solutions to overcome above distribution system measurement problems
- Design the prototype of a µPMU with required specifications to be used in distribution system
- Conduct research on areas of applicability of µPMU data

1.4 Overview

The project initially conducts a study on the difference between transmission and distribution systems which do not let us use the same Phasor Measurement Unit for both of them. It is followed by a research on the specifications needed for a measurement unit to be employed in distribution system. Based on the requirements, solutions to cater the distribution system problems are introduced. The specifications for the prototype of a μPMU are decided taking into hardware realization constraints into account. A Microcontroller development kit from Texas Instruments is used for analog signal acquisition, signal processing and Phasor Estimation. Time information for time-stamping the signal is retrieved from the internet using an ESP 8266 NodeMCU Wifi Development Module. It mimics the system where GPS signal is centrally received and distributed to several PMUs in the area with limited error resolution. Time information is conveyed to the microcontroller via UART communication and hence time stamping is done. Then the results out of this prototype is analysed and compared with ideal case. At last, applications of μPMU data is examined and is presented.

Chapter 2

Micro- PMU Specifications and Prototype Development

2.1 Solutions to Distribution System Problems

Problem 1: Resolution of Distribution System Measurements

The voltage angle differences between locations on a distribution circuit will be up to two orders of magnitude smaller than those on a transmission network. An estimate of voltage and phase angle magnitude shows:

Resolution required for distribution systems are:

- Voltage Angle resolution for a standard distribution system of 33 kV, is 1V
- Phase angle resolution (Angle between two buses) is as low as 0.02^o

Taking the possible measurement error sources and above specifications into account, the specifications of an Ideal Micro-PMU is decided.

Solution for Problem 1: Specifying µPMU parameters to achieve above accuracy

The µPMU parameters are set accordingly as to achieve the above accuracy. Sample calculation of above parameters is done below:

Calculation of ADC Bit Width:

The Bit width of Analog to Digital Converter to be used in µPMU has to be specified. This is calculated from the requirement of voltage angle resolution.

Voltage angle resolution = 1V for 33 kV system

Vref = 33 kV

Vmin = 1V

Bit width = $\log_2\left(\frac{33000}{1}\right) \sim 16$

So, **ADC Bit width must be greater than or equal to 16** for healthier operation in distribution systems.

Calculation of Sampling Frequency:

The sampling frequency is calculated based on the requirement of phase angle resolution.

Phase angle resolution for distribution systems = 0.02^o

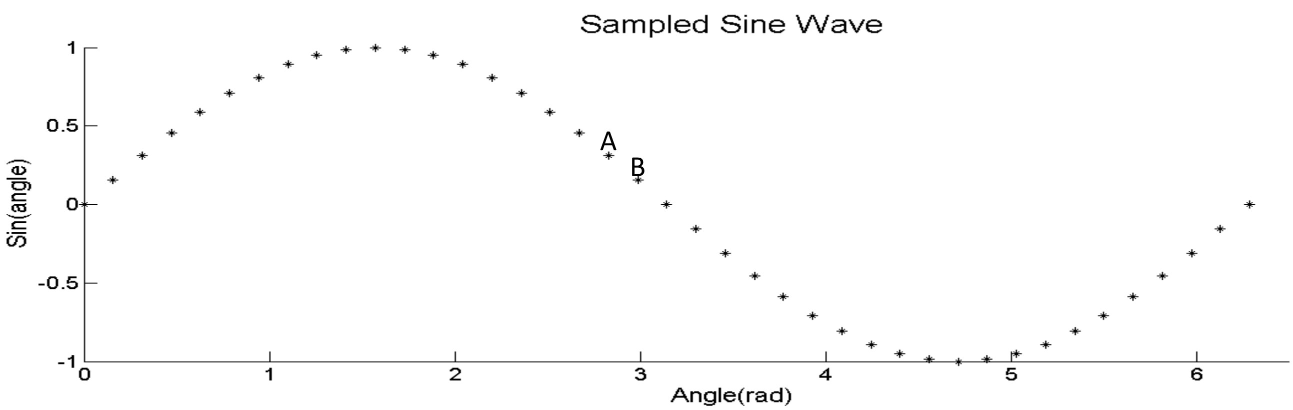


Figure 2.1 Sampled Sine Wave

In the above figure, for the points A and B to have angular difference equal to the angular resolution in distribution systems (=0.02^o), we must have sampling period maximum of 0.02^o

Lets say, Sampling provides N samples/cycle

0.02^o corresponds to 1 sample acquisition

360^o corresponds to $\frac{1}{0.02} * 360 = 18000$ samples acquisition

Hence the **Sampling Frequency must be at least 18 kHz**.

Calculation of System Clock Frequency:

The system clock rate is decided from the limitation on estimation of peak of sinusoid during phasor estimation process. In PMU, the reported phase angle is the angle between the positive peak and the reference GPS time in the signal cycle. For this, the time at which the peak is occurring for the signal must be estimated. Let's say, two samples A and B (given in Fig. 2.2) are sensed, between which two samples, the signal peak can be expected to lie. Here, we apply some efficient extrapolation technique and mention that the peak is x amount of clock cycles away from the 1^{st} sample.

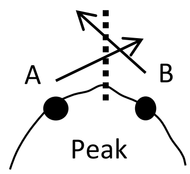


Figure 2.2 Peak Detection Technique using Extrapolation

For distribution systems, the error in the peak is limited to 1 milidegree.

So, 0.001^{o} must be expressed as an integer no. of clock cycles. As this is the least count, 0.001^{o} corresponds to 1 clock cycle.

360^{o} corresponds to 0.02 seconds

0.001^{o} corresponds to: $\frac{0.02}{360} * 0.001 = \frac{0.00002}{360}\ seconds$

So, clock frequency = $\frac{1}{Time\ Period} = \frac{360}{0.00002} = 18\ MHz$

So, **System Clock Frequency must be a minimum of 18 MHz** for less erratic operation.

So, from the above analysis, we conclude that the specifications for an ideal μPMU must be the following:

Parameter	Parameter Limits	Parameter Limited by
ADC Bit Width	>16	Voltage Magnitude Resolution
ADC Sampling Frequency	>18 kHz	Phase Angle Resolution
System Clock Frequency	>18MHz	Limit on error in peak detection
GPS Lock Error	<1 μS	Limit on position of GPS Reference Signal

Table 2.1 Ideal μPMU Parameter Limits

Problem 2: Significant Error due to System Clock Drift

In case of distribution systems, significant amount of error can be introduced into the phasors due to system clock drift. To cater this, an innovative idea of compensating the time stamp with consideration of system clock drift is introduced. There will be a Predictive Compensator near the phase locked oscillator block in fig. 1.1 which will predict the amount of system clock drift from GPS time and will compensate it while time stamping. The variation of system time from GPS time for positive and negative drifts can be plotted as given in Fig. 2.3.

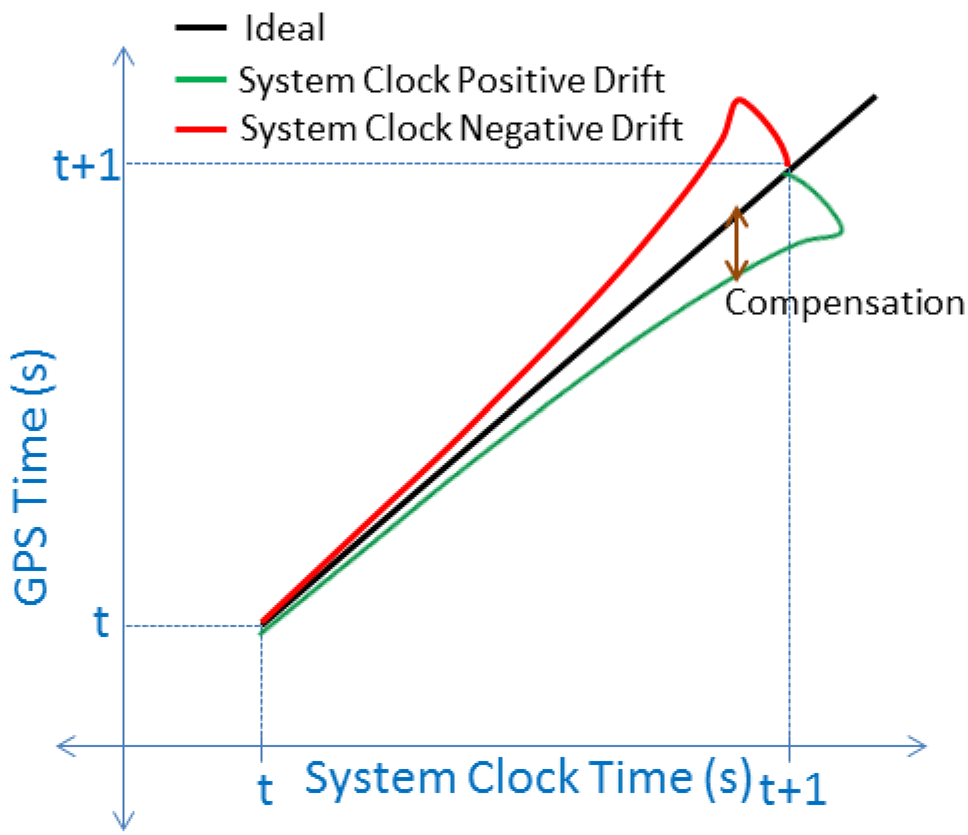


Figure 2.3 Variation of System Time with GPS Time with positive and negative clock drift

2.2 Micro- PMU Prototype Parameter Specifications

Based on ratings for an ideal µPMU given in table 2.1, a prototype of a standard µPMU was prepared using Microcontroller Development Kit from Texas Instruments (for Voltage/Current Signal Acquision, Processing and Phasor Estimation) and ESP 8266 Wifi Development Board (for Time Signal Acquisition from internet and synchronization). Based on hardware and software optimization limitations, the ratings for the prototype of the µPMU was choosen to be the following (Table 2.2)

Parameter	Parameter Value
ADC Bit Width	12
ADC Sampling Frequency	2.5 kHz
System Clock Frequency	16 MHz
Reporting Rate	50 Hz

Table 2.2 µPMU Parameter Specifications

2.3 Hardware Requirement for Micro- PMU Prototype Realization

The following hardwares were required for realization of prototype of a standard μPMU:

- Tiva^{TM} C Series TM4C123G Launch Pad (EK- TM4C123GXL) from Texas Instruments
  This microcontroller was used for Voltage/Current Signal Acquision, Processing and Phasor Estimation.

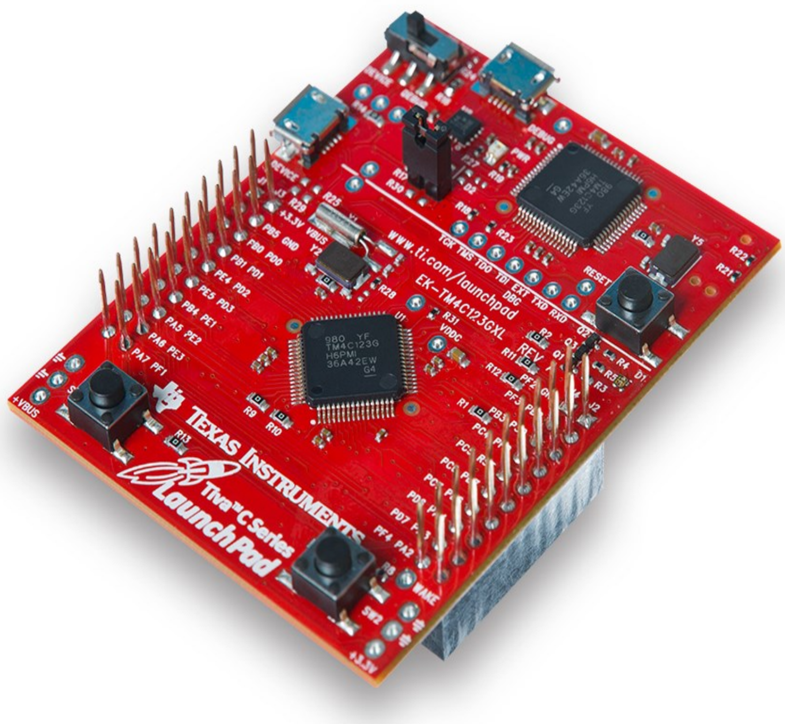


Figure 2.4 Tiva^{TM} C Series TM4C123G Launch Pad (EK- TM4C123GXL)

The specifications of this microcontroller is the following:

A. 32-bit ARM® Cortex™-M4F architecture
B. Clock Frequency: 16 MHz (Programmable upto 80 MHz)
C. 12 Bit ADC
D. Motion control PWM

More details about this microcontroller is given in Appendix 1.

• ESP 8266 Node Mcu Wifi Development Board (Shown in Fig. 2.5)

Figure 2.5 ESP 8266 Node Mcu Wifi Development Board

2.4 Software Requirement for Micro- PMU Prototype Realization

Software to program TM4C123G Microcontroller: IAR Embedded Workbench from IAR Systems

Software to program ESP 8266 Wifi Development Board: Arduino IDE from Arduino

2.5 Circuit Diagram

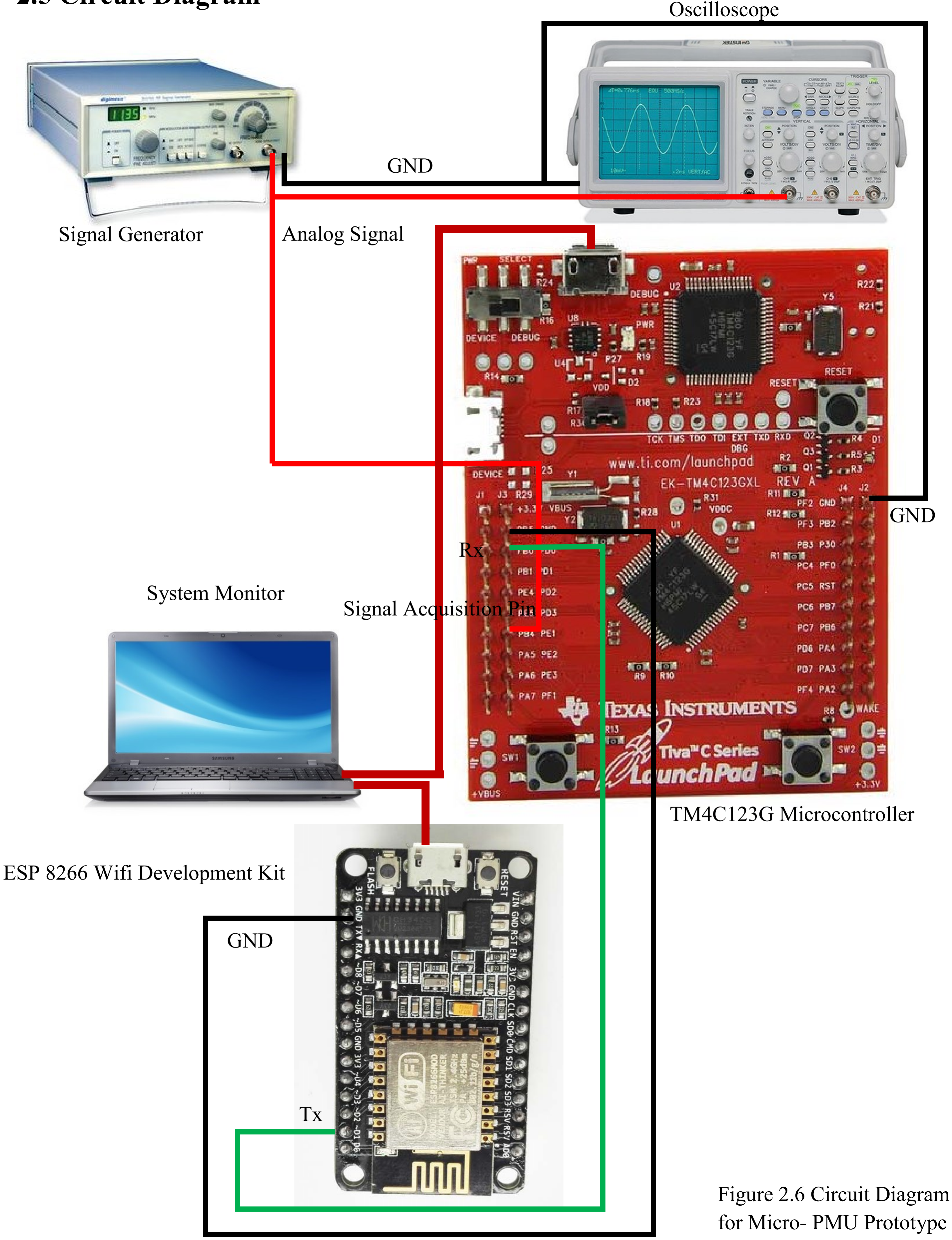
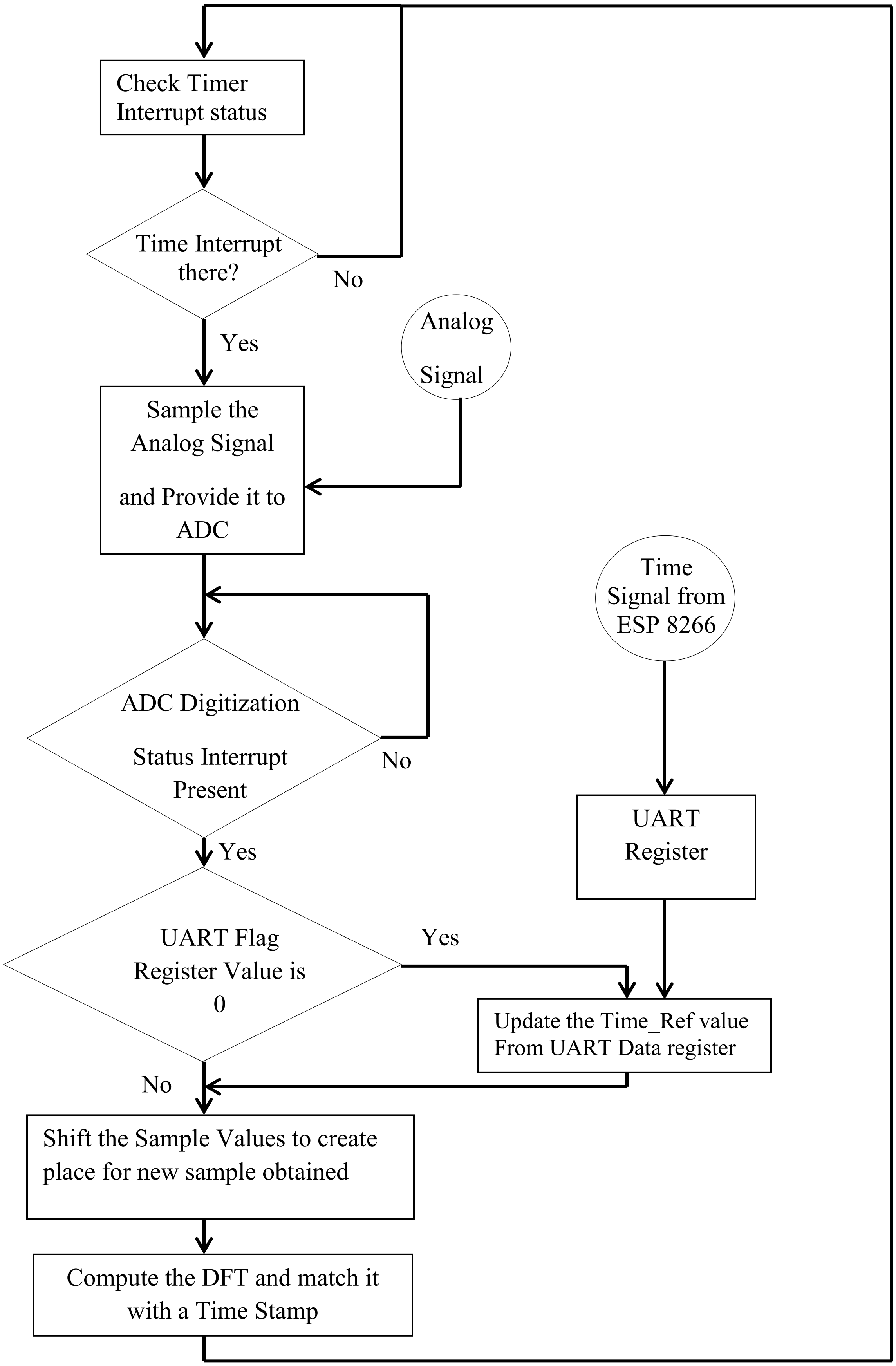


Figure 2.6 Circuit Diagram for Micro- PMU Prototype

2.6 Description of events in TM4C123G Microcontroller

2.6.1 Flowchart



2.6.2 Block Diagram of Operations inside TM4C123G Microcontroller

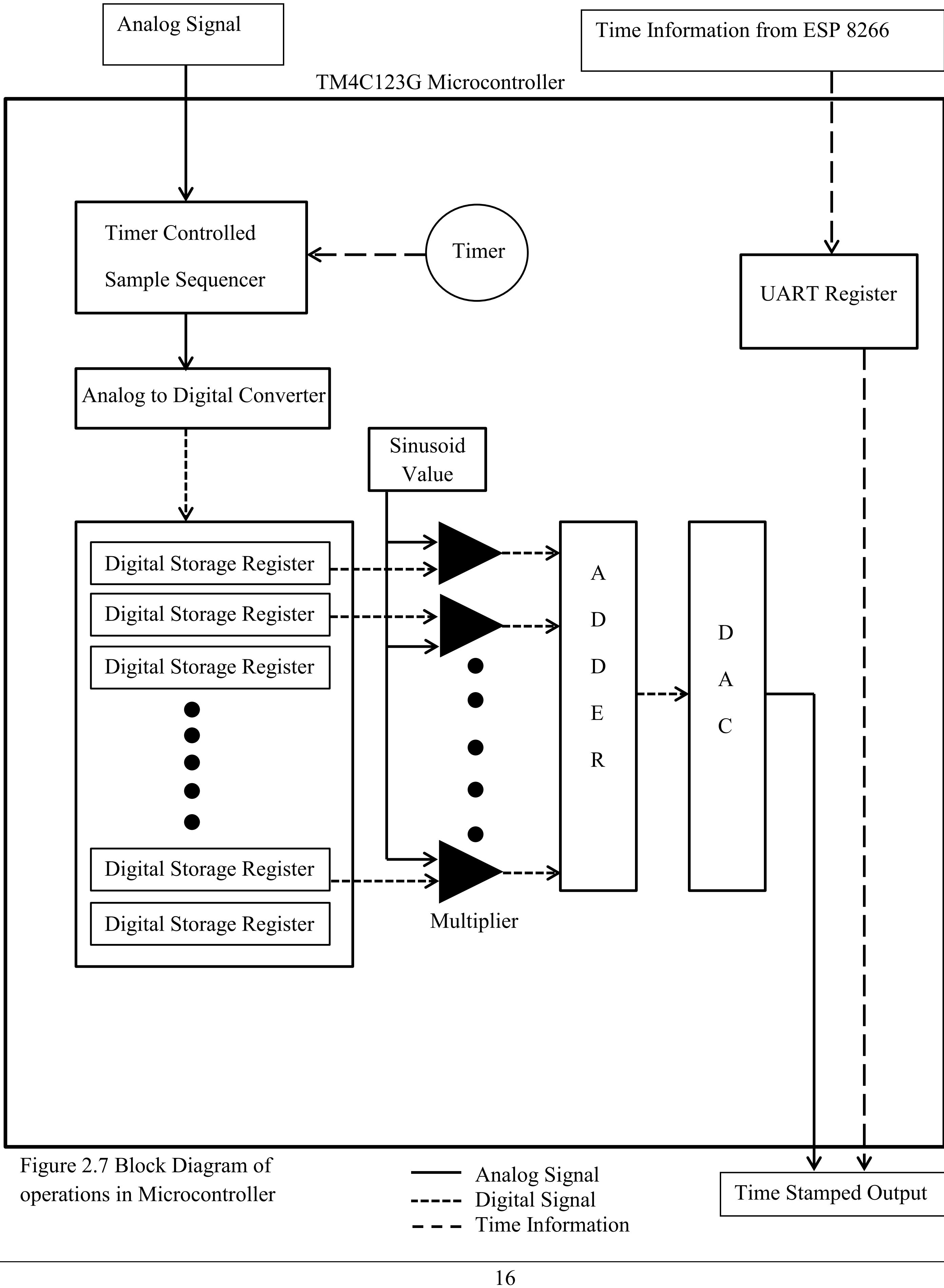


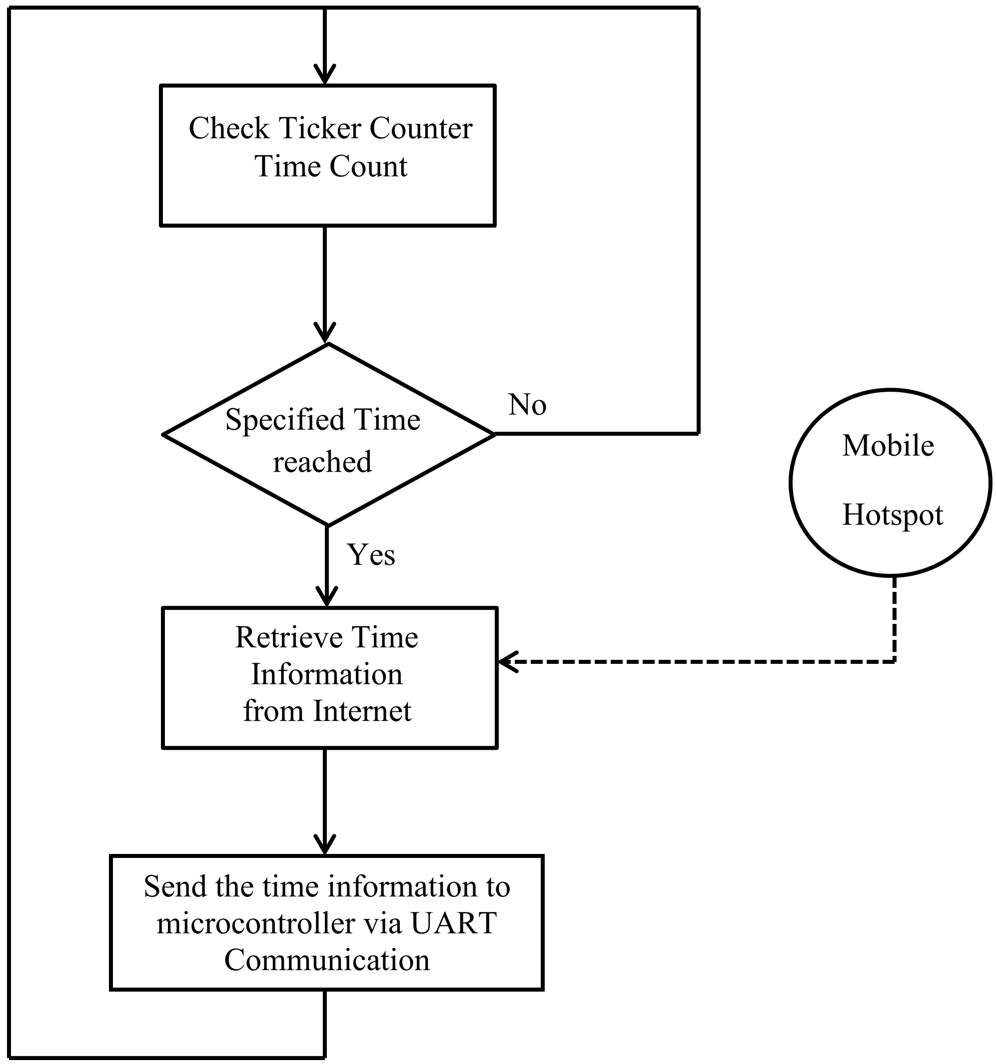
Figure 2.7 Block Diagram of operations in Microcontroller

2.6.3 Description of events in TM4C123G Microcontroller

The TM4C123G Microcontroller gets analog signals a from a signal generator. The reference voltage is 3.3 V for Microcontroller, so we apply a sinusoidal signal of 1.65 V magnitude with a DC offset of 1.65 V. TM4C123G comes inbuilt with 2 ADCs of Successive Approximation Register (SAR) type. The ADCs are of 12 bits each. The sampling of the ADC can be both continuous or at some particular sampling frequency. The microcontroller has 4 sample sequencers which have the capacity to sample upto 8 Analog channels at a time. As we have only one analog signal, we are using Sample Sequencer 0 which samples one analog input channel. The sample sequencer is configured to sample at 2.5 kHz using time count from timer. The ADC digitizes the signal and sends an interrupt upon successful conversion. We continuously monitor the status of Flag of UART register. The flag becomes '0' when it receives a signal through UART communication. Upon signal reception as sensed by the UART register flag, we update the reference time based on the time data in the data register of UART. We also continuously monitor the status of ADC sample conversion interrupt. If there is an interrupt, we run the DFT computation loop. Initially the newly acquired value needs to be stored and the oldest sample in DFT window must be discarded. Then the DFT calculation is done and we get the real and imaginary part of the phasor. This data is time stamped with the current time (Time reference + number of clock cycles passed after time signal was acquired). Then the time stamped data is sent to monitoring system for further communication to PDC.

2.7 Description of events in ESP 8266 Wifi Module

2.7.1 Flowchart



2.7.2 Block Diagram of Operations in ESP 8266 Wifi Module

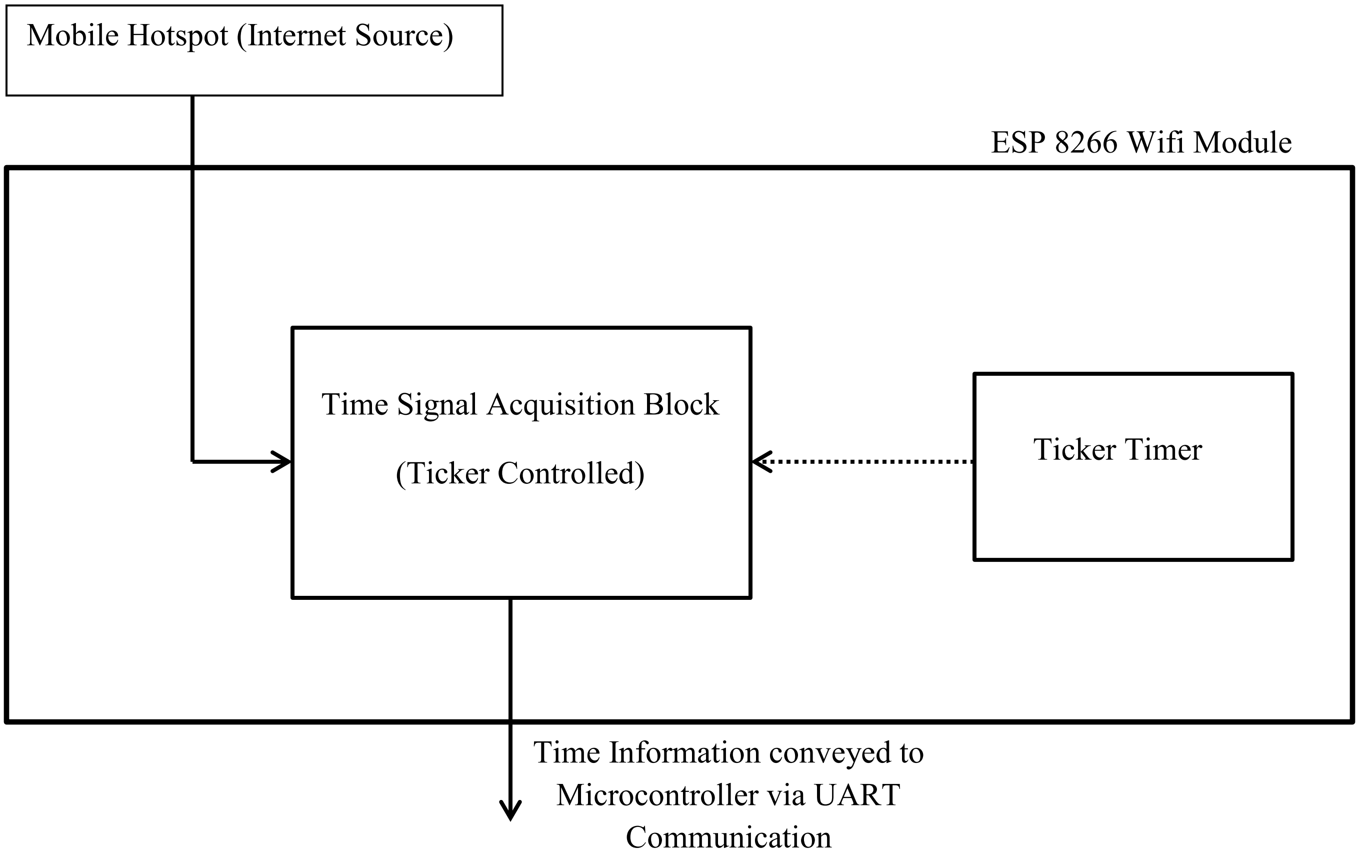


Figure 2.8 Block Diagram of Operations in Wifi Module

2.7.3 Description of Operations in ESP 8266 Wifi Module

The Wifi module is programmed in Arduino software. Arduino has a function named Ticker function which repeats a particular function again and again after a specified amount of time. We use the same function to demand time information from internet. Here, the internet source is hotspot created in user's mobile. The time signal so acquired is sent to the microcontroller using UART Communication and the whole process is repeated again and again at specified intervals specified by the ticker function.

2.8 UART Communication

The universal asynchronous receiver/transmitter (UART) takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. [9] Each UART contains a shift register which is the fundamental method of conversion between serial and parallel forms. Serial transmission of digital information (bits) through a single wire or other medium is less costly than parallel transmission through multiple wires. The UART Communication can be pictorially shown as given in Fig. 2.9.

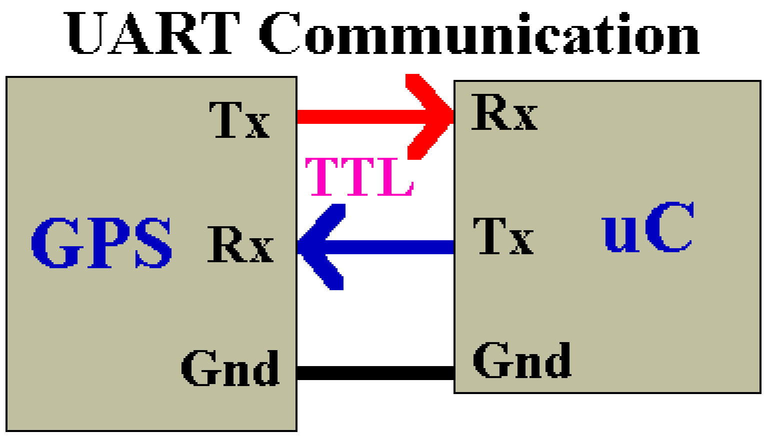


Figure 2.9 Block Diagram of UART Communication

UART Communication Protocol is used to transmit time information from ESP 8266 Module to TM4C123G microcontroller.

2.9 Data Transmission Capabilities

A μPMU is incomplete without provisions for data transmission to PDCs. In the prototype prepared the data transmission capability to the μPMU is provided. The time stamped phasor data so computed by the microcontroller is achieved at the monitoring system (Laptop or a PC). This data can be communicated to further levels through LAN/Ethernet or some other mode of communication adhering to standard C37.118 protocol for PMUs.

Chapter 3

Results

The μPMU was tested with 1.65 V signal from a signal generator with continuous windowing mode. The DFT window changes with every new sample acquired. In this case, the window at an instant looks like the one given in Fig. 3.1.

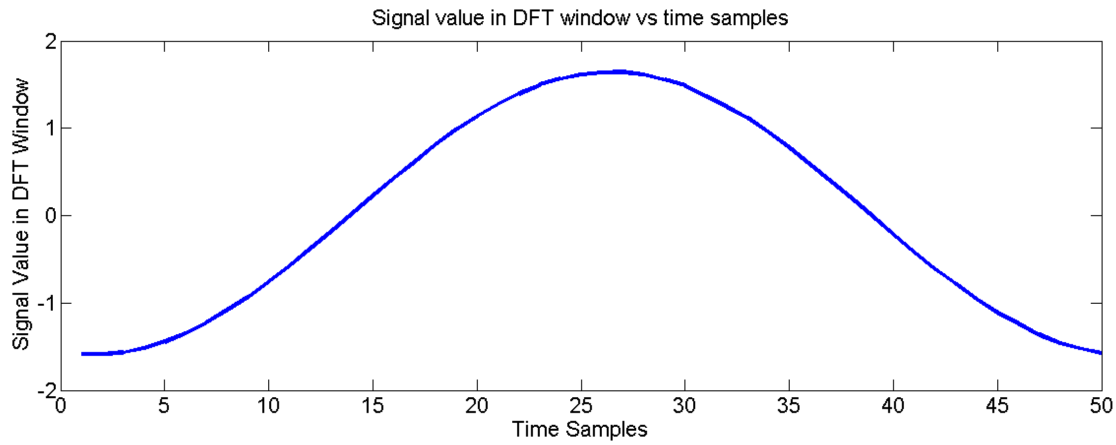


Figure 3.1 Window of DFT at an instant

The magnitude and phase plot of phasor with respect to time samples is given in Fig.3.2 and Fig.3.3 respectively.

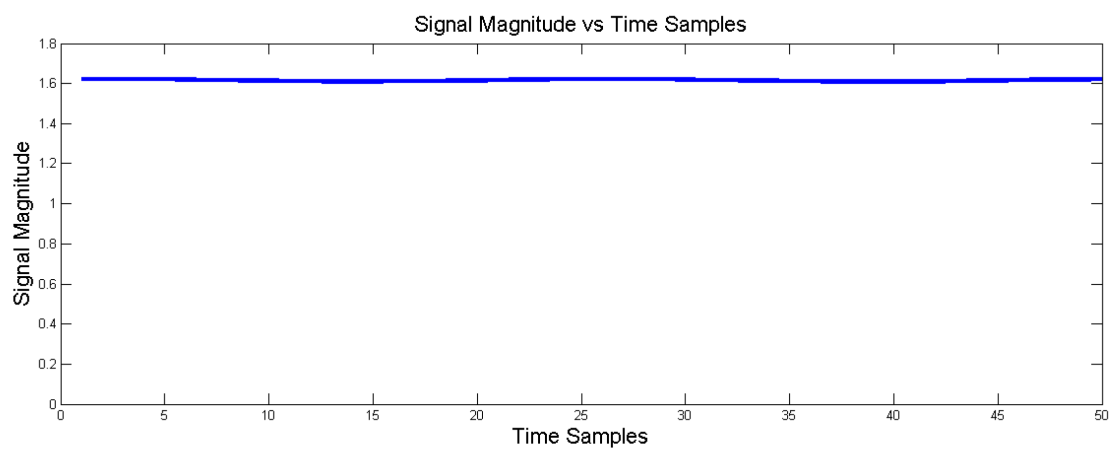


Figure 3.2 Signal Magnitude vs Time samples

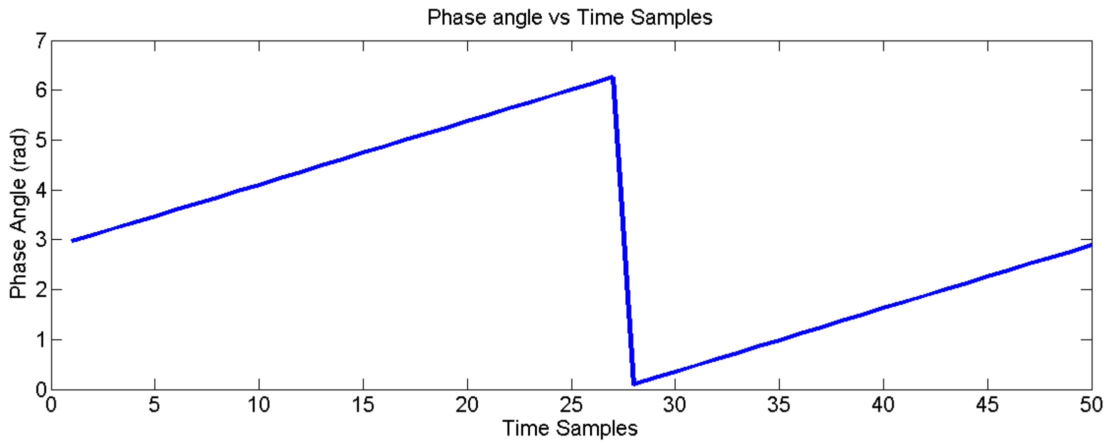


Figure 3.3 Phase angle vs time samples

The results are coming as expected. For a cycle, the magnitude of phasor remains the same but the phase varies from 0 to 2∏ (=6.28..)

Chapter 4

Applications of Micro- PMU data

The applications of μPMU data can be divided into two types, Diagnostic Applications and Control Applications.

A. Diagnostic Applications

1) Unintentional island detection

The objective is to quickly and reliably recognize a potentially unsafe situation where a set of distributed generation (DG) generators and loads have separated from the grid but continue to energize their local portion of the network. Today's inverters have very reliable anti-islanding protection. However, with greater penetration of diverse distributed resources and more complex dynamics on distribution circuits, it may become increasingly difficult to distinguish fault events from other abnormal conditions where it is desirable to keep DG online (for example, low-voltage ride-through).

2) Topology status verification

The objective is to detect or confirm the actual status (open or closed) of field switches whose indicators may be unavailable remotely or considered unreliable. Knowledge of the network topology is essential to inform safe operations and accurate estimation of the system state.

3) Phase identification and balancing

The simple yet important function here is to identify the connection of single-phase loads and laterals to phases A, B or C, to facilitate proper balancing. Direct phase angle measurement with a portable device on the secondary distribution system could be a uniquely quick and easy way to ascertain this.

4) Reverse power flow detection

The goal is to identify, or rather anticipate, when power flows in reverse direction on a radial distribution feeder. While reverse flow may be unproblematic in some situations, its significance depends on the type of protection system design used, and whether the coordination of protective devices could be compromised under reverse flow conditions [10]. Voltage regulation may also be impacted by reverse flow, if control systems are designed based on the assumption of a declining voltage profile toward the end of the feeder.

5) State estimation

State estimation means identifying as closely as possible, from available network models and empirical measurements, the operating state of the a.c. system in near real-time.[11] This state is completely described if the two state variables that drive real and reactive power flow – namely, voltage magnitude and phase angle – are known or computed for every node in the network, given connectivity and impedances of network branches. State estimation is generally more difficult for distribution than for transmission systems. This is because distribution systems are harder to model (owing to untransposed lines with phase imbalances, small X/R ratios, large numbers of connecting load points, and less redundancy from Kirchhoff's laws) and present a high-dimensional mathematical problem, while at the same time offering few physical measurements to inform the state estimation. Data from µPMUs could explicitly provide these state variables to directly feed into a Distributed State Estimator (DSE), which in turn may provide information to a Distribution Management System (DMS).

6) Fault location

The goal is to infer the actual geographical location of a fault on a distribution feeder to within a small circuit section (compared to the distance between protective devices) by using recorded measurements of voltage angle before and during the fault, and interpreting these in the context of a circuit model. Algorithms exist for locating faults through proper analysis of monitored data, but the quality of available measurements on distribution circuits is often insufficient to support them. We expect that voltage angle might enable fault location with greater precision than before.

B. Control Applications

Beyond enhanced diagnostic capabilities, synchrophasor data may enable more refined management and active control of distribution systems. Possible control applications include the following:

1) Protective Relaying

Reverse power flow was noted above as a condition that can be important to diagnose and avoid, but another approach is to employ protection schemes that safely accommodate reverse flow. Without requiring a costly replacement of protective devices, it may be feasible to develop supervisory differential relaying schemes based on µPMU data that recommend settings to individual devices based on overall system conditions, which might include reverse flow. This approach is being demonstrated and tested at the transmission level in a DOE-funded Adaptive Relaying project [12].

2) Volt-VAR Optimization

We do not expect that voltage angle measurement would afford an inherent advantage over magnitude for feeder voltage optimization, but the capability to support this important function alongside other applications could add significantly to the business case.

3) Microgrid Coordination

To advance the opportunities for active control based on μPMU measurements, we should study requirements for hierarchical, layered, distributed control of an islandable cluster of aggregated distributed resources and identify the merits, if any, of angle as a state variable. Microgrid balancing and synchronization is an application with a longer strategic time horizon, but one where the use of voltage angle as a control variable is expected to be crucial.

Chapter 5

Conclusions and Scope for Future Work

The project presented a prototype of a standard micro- PMU with specifications well within acceptable range of values and also fitting best with hardware implementation limits. The parameters such as ADC Bit width, sampling frequency, system clock frequency, System clock drift from GPS time, GPS drift etc. decide the level to which the phasor estimation and time stamping process is accurate. For varying levels of voltages, ADC cascading can be looked into. Improvements Also Field Programmable Gate Array (FPGA) processors, which provide the ability to use any number of bits and allow user definition of hardware specifications, can be looked into for hardware optimization. With more and more amount of optimization in all the computation process involved in the µPMU operation, better and better levels of accuracy can be obtained through increase in sampling rates. With efficient methods of compensation, time stamp for a phasor can be decided with much precision. Advancement in Embedded System technology and invention of powerful microcontrollers also lead to higher and higher accuracy levels synchrophasor technology.

The distribution system is becoming much complex with inclusion of distributed energy sources. Hence the number of µPMUs to be placed and the location of µPMU placement must be optimized. Proper algorithms for µPMU placements must be thought of so as to provide better business case for µPMU installation.

Appendices

Appendix 1

Tiva™ C Series TM4C123G Launch Pad (EK- TM4C123GXL) from Texas Instruments- Technical Information

Feature	Description
Performance	
Core	ARM Cortex-M4F processor core
Performance	80-MHz operation; 100 DMIPS performance
Flash	256 KB single-cycle Flash memory
System SRAM	32 KB single-cycle SRAM
EEPROM	2KB of EEPROM
Internal ROM	Internal ROM loaded with TivaWare™ for C Series software
Security	
Communication Interfaces	
Universal Asynchronous Receivers/Transmitter (UART)	Eight UARTs
Synchronous Serial Interface (SSI)	Four SSI modules
Inter-Integrated Circuit (I^2C)	Four I^2C modules with four transmission speeds including high-speed mode
Controller Area Network (CAN)	Two CAN 2.0 A/B controllers
Universal Serial Bus (USB)	USB 2.0 OTG/Host/Device
System Integration	
Micro Direct Memory Access (µDMA)	ARM® PrimeCell® 32-channel configurable µDMA controller
General-Purpose Timer (GPTM)	Six 16/32-bit GPTM blocks and six 32/64-bit Wide GPTM blocks
Watchdog Timer (WDT)	Two watchdog timers
Hibernation Module (HIB)	Low-power battery-backed Hibernation module
General-Purpose Input/Output (GPIO)	Six physical GPIO blocks
Advanced Motion Control	
Pulse Width Modulator (PWM)	Two PWM modules, each with four PWM generator blocks and a control block, for a total of 16 PWM outputs.
Quadrature Encoder Interface (QEI)	Two QEI modules
Analog Support	
Analog-to-Digital Converter (ADC)	Two 12-bit ADC modules, each with a maximum sample rate of one million samples/second
Analog Comparator Controller	Two independent integrated analog comparators
Digital Comparator	16 digital comparators
JTAG and Serial Wire Debug (SWD)	One JTAG module with integrated ARM SWD

Appendix 2

Code for TM4C123G Microcontroller

```
#include <lm4f120h5qr.h>
#include <arm_math.h>
#include <math.h>

#define ADC_FLAG_NONE 0x00000000
#define ADC_FLAG_TOGGLE_LED 0x00000001

char readChar(void);
int get_time_ref();

volatile static float32_t adcResult[50];

//float32_t sin_sample,cos_sample;
float32_t sine_sum,cosine_sum,MAG,PHG=0;
float32_t IMAG[50],REAL[50];
int time_ref,TIME_STAMP[50];
int constant,reference,count[50];
float32_t sin_multiple[50],cos_multiple[50];

volatile static uint8_t adc_flag = ADC_FLAG_NONE;

void ADC1SS3_Handler(void){
  adc_flag = ADC_FLAG_TOGGLE_LED;
  ADC1->ISC = (1<<3);
}

void TIMER0A_Handler(void){
  TIMER0->ICR |= (1<<0);
}

int main()
{
  int i;
  for(i=0;i<50;i++)sin_multiple[i] = arm_sin_f32(2*PI*(49-i)/50);
  for(i=0;i<50;i++)cos_multiple[i] = arm_cos_f32(2*PI*(49-i)/50);

  //CONFIGURE THE UART
  SYSCTL->RCGCUART |= (1<<1);
  SYSCTL->RCGCGPIO |= (1<<1);
  GPIOB->AFSEL = (1<<1)|(1<<0);
  GPIOB->PCTL = (1<<0)|(1<<4);
  GPIOB->DEN = (1<<0)|(1<<1);

  UART1->CTL &= ~(1<<0);
  UART1->IBRD = 104;
  UART1->FBRD = 11;
  UART1->LCRH = (0x3<<5);
  UART1->CC = 0x0;
  UART1->CTL = (1<<0)|(1<<8)|(1<<9);

  //CONFIGURE GPIOF LED
  SYSCTL->RCGCGPIO |= (1<<5);
  GPIOF->DEN = 0xff;
```

```
  GPIOF->AFSEL = 0x00;
  GPIOF->DIR = 0xff;
  GPIOF->DATA &= ~((1<<1)|(1<<2)|(1<<3));

  //CONFIGURE THE ADC
  SYSCTL->RCGCADC = (1<<1);//ENABLE CLOCK TO RCGCADC REGISTER-WE ARE SETTING
BIT 1 BECAUSE WE WILL USE ADC1 NOT ADC0
  SYSCTL->RCGCGPIO |= (1<<4);//ENABLE PORT E, SO SET BIT 4
  GPIOE->DIR &= ~(1<<1);// WE WILL USE PIN 1, PE1 AS INPUT SO WE NEED TO CLEAR
IT
  GPIOE->AFSEL = (1<<1);//which pin using for analog input, here PE1,ALTERNATE
FUNCTION SELECT(AFSEL)
  GPIOE->DEN &= ~(1<<1);//DEN bit (Digital Enable bit) cleared to activate the
analog input pin, as PE1 is analog
  GPIOE->AMSEL = (1<<1);//disable analog isolation circuit

  //CONFIGURE THE SAMPLE SEQUENCER
  ADC1->ACTSS &= ~(1<<3);//first disable sequencer before configuring, as we
are choosing SS3, 1<<3 came
  ADC1->EMUX = ((0x5)<<12);//Difference//Configure the trigger what starts the
conversion, how do u tell the microcontroller to begin reading and then
converting
  ADC1->SSMUX3 = 2;//which analog input you are choosing, as we are using PE1,
it is AN2, so 2 used
  ADC1->SSCTL3 = 0x6;//HAVE TO TAKE CARE TO SET THIS
  ADC1->IM = (1<<3);//as we are using SS3, 1<<3
  ADC1->ACTSS |= (1<<3);//previously we disabled the sequencer, now we need to
enable it

  //SET THE ADC INTERRUPT SIGNAL
  ADC1->ISC = (1<<3);//clear the interrupt flags
  NVIC_EnableIRQ(ADC1SS3_IRQn);//ADC1SS3_IRQn is an interrupt specified in the
lm4f120h5qr header, we are including it in cstartup_M file


  //CONFIGURE THE TIMER
  SYSCTL->RCGCTIMER |= (1<<0);//ENABLE CLOCK TO RCGCTIMER REGISTER
  TIMER0->CTL &= ~(1<<0);//ENSURE TIMER IS DISABLED BEFORE MAKING ANY CHANGES
  TIMER0->CFG |= 0x00000000;//GPM CONFIGURATION REGISTER
  TIMER0->TAMR |= ((0x2)<<0);//WE ARE USING TIMER A so tAmr(There are two type
of timers, A and B, within Timer A, Timer0,1 are there)
                              //0x2 is for periodic mode of timer operation
  TIMER0->TAMR &= ~(1<<4);//WE ARE PUTTING O IN BIT 4 SO THAT THE TIMER
DIRECTION IS COUNT-DOWN NOT COUNT-UP
  TIMER0->TAILR = 0x1900;//AS WE HAVE DOWN COUNTER, WE NEED TO SET THE STARTING
VALUE OF COUNTING, HERE IT IS 16,000,000 (ONE CLOCK CYCLE)
  TIMER0->CTL |= (1<<5);//THIS IS DONE TO ENABLE OUTPUT TIMER A ADC TRIGGER

  char c[10];
  int val[10];
  constant = 0;
  while((UART1->FR & (1<<4)) != 0);
  c[0] = UART1->DR;
  val[0] = c[0];
  constant = constant + val[0] - 48;
  for(int i=1;i<10;i++)
  {
     GPIOF->DATA = (1<<((i%3)+1));
     c[i] = readChar();
     val[i] = c[i];
```

```
      if(i<7)constant = constant + (val[i]-48)*pow(10,i);
    }
    reference = 0;

    //SET THE TIMER INTERRUPT SIGNAL
    TIMER0->IMR |= (1<<0);//INTERRUPT IS ENABLED
    NVIC_EnableIRQ(TIMER0A_IRQn);//INTERRUPT FUNCTION
    TIMER0->CTL |= (1<<0);//NOW WE ENABLE THE TIMER

    while(1){
      if((UART1->FR & (1<<4)) == 0)
      {
        time_ref = get_time_ref();
        reference = 0;
      }
      if(adc_flag == ADC_FLAG_TOGGLE_LED){
        //SHIFT THE adcResult VALUES AND KEEP UPDATED VALUE IN 0TH PLACE
        cosine_sum = cosine_sum - (adcResult[49]/25);
        for(i=49;i>0;i--){
          adcResult[i] = adcResult[i-1];
        }
        adcResult[0] = (((ADC1->SSFIFO3)*3.3/4096)-1.65);
        cosine_sum = 0;
        sine_sum = 0;
        for(i=0;i<50;i++)
        {
          cosine_sum = cosine_sum + adcResult[i]*cos_multiple[i];
          sine_sum = sine_sum + adcResult[i]*sin_multiple[i];
        }

        for(i=49;i>0;i--){
          IMAG[i] = IMAG[i-1];
          REAL[i] = REAL[i-1];
          TIME_STAMP[i] = TIME_STAMP[i-1];
          count[i] = count[i-1];
        }
        IMAG[0] = sine_sum;
        REAL[0] = cosine_sum;
        TIME_STAMP[0] = time_ref;
        count[0] = reference;

        reference++;
        adc_flag = ADC_FLAG_NONE;
      }
    }
    return 0;
  }

  int get_time_ref()
  {
    char c[10];
    int val[10];
    int time_ref = 0;
    for(int i=0;i<10;i++)
    {
      GPIOF->DATA = (1<<((i%3)+1));
      c[i] = readChar();
      val[i] = c[i];
      if(i>5)time_ref = time_ref + (val[i]-48)*pow(10,9-i);
    }
```

```
  return time_ref;
}

char readChar(void)
{
  char c;
  while((UART1->FR & (1<<4)) != 0);
  c = UART1->DR;
  return c;
}
```

Interrupt Handler Code

```
#pragma language=extended
#pragma segment="CSTACK"

extern void __iar_program_start( void );

extern void NMI_Handler( void );
extern void HardFault_Handler( void );
extern void MemManage_Handler( void );
extern void BusFault_Handler( void );
extern void UsageFault_Handler( void );
extern void SVC_Handler( void );
extern void DebugMon_Handler( void );
extern void PendSV_Handler( void );
extern void SysTick_Handler( void );
extern void TIMER0A_Handler(void);
extern void ADC1SS3_Handler(void);

typedef void( *intfunc )( void );
typedef union { intfunc __fun; void * __ptr; } intvec_elem;

// The vector table is normally located at address 0.
// When debugging in RAM, it can be located in RAM, aligned to at least 2^6.
// If you need to define interrupt service routines,
// make a copy of this file and include it in your project.
// The name "__vector_table" has special meaning for C-SPY, which
// is where to find the SP start value.
// If vector table is not located at address 0, the user has to initialize
// the  NVIC vector table register (VTOR) before using interrupts.


#pragma location = ".intvec"
const intvec_elem __vector_table[] =
{
  { .__ptr = __sfe( "CSTACK" ) },
  __iar_program_start,

  NMI_Handler,
  HardFault_Handler,
  MemManage_Handler,
  BusFault_Handler,
  UsageFault_Handler,
  0,
  0,
  0,
  0,
  SVC_Handler,
```

```
  DebugMon_Handler,
  0,
  PendSV_Handler,
  SysTick_Handler,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  TIMER0A_Handler,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,//14
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  ADC1SS3_Handler
};

#pragma call_graph_root = "interrupt"
__weak void NMI_Handler( void ) { while (1) {} }
```

```
#pragma call_graph_root = "interrupt"
__weak void HardFault_Handler( void ) { while (1) {} }
#pragma call_graph_root = "interrupt"
__weak void MemManage_Handler( void ) { while (1) {} }
#pragma call_graph_root = "interrupt"
__weak void BusFault_Handler( void ) { while (1) {} }
#pragma call_graph_root = "interrupt"
__weak void UsageFault_Handler( void ) { while (1) {} }
#pragma call_graph_root = "interrupt"
__weak void SVC_Handler( void ) { while (1) {} }
#pragma call_graph_root = "interrupt"
__weak void DebugMon_Handler( void ) { while (1) {} }
#pragma call_graph_root = "interrupt"
__weak void PendSV_Handler( void ) { while (1) {} }
#pragma call_graph_root = "interrupt"
__weak void SysTick_Handler( void ) { while (1) {} }
#pragma call_graph_root = "interrupt"
__weak void TIMER0A_Handler( void ) { while (1) {} }
#pragma call_graph_root = "interrupt"
__weak void ADC1SS3_Handler( void ) { while (1) {} }


void __cmain( void );
__weak void __iar_init_core( void );
__weak void __iar_init_vfp( void );

#pragma required=__vector_table
void __iar_program_start( void )
{
  __iar_init_core();
  __iar_init_vfp();
  __cmain();
}
```

Appendix 3

Code for ESP 8266 Wifi Development Module

```
#include <Arduino.h>

#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>

#include <ESP8266HTTPClient.h>
#include <SoftwareSerial.h>
#include <Ticker.h>

Ticker ticker;

boolean ticker_reached;
boolean LED_state;

void ticker_handler(){
  ticker_reached = true;
}

#define USE_SERIAL Serial
```

```
ESP8266WiFiMulti WiFiMulti;

SoftwareSerial swser(4,5);

void setup() {

    pinMode(LED_BUILTIN, OUTPUT);

    ticker_reached = false;
    LED_state = HIGH;

    //call ticker_handler() in 5 second
    ticker.attach(5, ticker_handler);

    swser.begin(9600);
    USE_SERIAL.begin(9600);
   // USE_SERIAL.setDebugOutput(true);

    USE_SERIAL.println();
    USE_SERIAL.println();
    USE_SERIAL.println();

    for(uint8_t t = 4; t > 0; t--) {
        USE_SERIAL.printf("[SETUP] WAIT %d...\n", t);
        USE_SERIAL.flush();
        delay(1000);
    }

    WiFiMulti.addAP("hari", "harijulu");

}

void loop() {
  if(ticker_reached){
    ticker_reached = false;
    digitalWrite(LED_BUILTIN, LED_state);
    LED_state = !LED_state;

    if((WiFiMulti.run() == WL_CONNECTED)) {

        HTTPClient http;

http.begin("api.thingspeak.com",80,"/apps/thinghttp/send_request?api_key=DM6CQ6
7HSJ39LF5Y"); //API


        int httpCode = http.GET();

        if(httpCode > 0) {
            if(httpCode == HTTP_CODE_OK) {
                String payload = http.getString();
                USE_SERIAL.println("Epoch Time Received, Sending data...");
                for(int i=0;i<10;i++)
                {
                  swser.print(payload[i]);
                  delay(50);
                }
            }
        }
        else {
```

```
            USE_SERIAL.printf("[HTTP] GET... failed, error: %s\n",
http.errorToString(httpCode).c_str());
        }

        http.end();
    }
    delay(500);
  }
}
```

References

[1] A. G. Phadke, "Synchronized phasor measurements-a historical overview," in *Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES*, 2002, pp. 476-479 vol.1.

[2] B. Pinte, M. Quinlan and K. Reinhard, "Low voltage micro-phasor measurement unit (μPMU)," *Power and Energy Conference at Illinois (PECI), 2015 IEEE*, Champaign, IL, 2015, pp. 1-4. doi: 10.1109/PECI.2015.7064888.

[3] A. von Meier, D. Culler, A. McEachern and R. Arghandeh, "Micro-synchrophasors for distribution systems," *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES*, Washington, DC, 2014, pp. 1-5.doi:10.1109/ ISGT.2014. 6816509.

[4] A.G. Phadke, J.Thorp and M.Adamiak, "A new measurement technique for tracking voltage phasors, local system frequency and rate of change of frequency", *IEEE Trans. Power App. Syst.,* Vol. PAS-102, no.5, pp. 1025-1038, May 1983

[5] "IEEE Standard for Synchrophasor for Power Systems", IEEE Std C37.118- 2005 *(Re-vision of IEEE Std. 1344-1995)*, pp. 1-57, 2006

[6] A.P. Meliopoulos, V.Madani, D. Novosel, G. Cokkinides, R.Alaileh, B. Fardanesh, et al., "Synchrophasor Measurement Accuracy Characterization", North American Synchrophasor Initiative Performance and Standards Task Team 2007.

[7] "TI GPS PPS Timing Application Note", Texas Instruments 2012.

[8] L. Hsiung Cheng, "Intelligent Neural Network- Based Fast Power System Harmonic Detection", *Industrial Electronics, IEEE Transactions on,* vol. 54, pp. 43-52, 2007.

[9] Y. y. Fang and X. j. Chen, "Design and simulation of UART serial communication module based on VHDL," *Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on*, Wuhan, 2011, pp. 1-4

[10] R. Arghandeh, R. Broadwater, "Interactive assessment of multi- harmonic sources in distribution networks", in Proc. 2012, ASME Power 2012 Conference

[11] C. N. Lu, J. H. Teng and W. H. E. Liu, "Distribution system state estimation," in *IEEE Transactions on Power Systems*, vol. 10, no. 1, pp. 229-240, Feb 1995.
doi: 10.1109/59.373946

[12] V. Centeno, R. King, and L. Cibulka. "Application of Advanced Wide Area Early Warning Systems with Adaptive Protection—Phase 2," CIEE , Project report to US DOE, Grant #DE-OE0000120, December 2011.