

IMPLEMENTASI PROTOKOL MODBUS TCP PADA SISTEM MONITORING BESARAN LISTRIK MENGGUNAKAN LABVIEW DAN POWER METER SCHNEIDER 810

Stefany Yunita Bara'langi¹, Ferdianto Tangdililing²

¹Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Atma Jaya Makassar
Alamat email : fbaralangi@lecturer.uajm.ac.id

²Program Studi Teknik Elektro, Fakultas Teknik, Universitas Atma Jaya Makassar
Alamat Email : ferdi82@rocketmail.com

ABSTRACT

This research aims to develop software applications that can perform data communication from a sensor electrical quantities to the Master Terminal Unit as a server with Modbus TCP protocol. Data communication architecture in real time from the sensor to the MTU so that the user can monitor the amount of electricity in a building at any time. Software applications that are built using tools LabView. The results of data acquisition can be stored in computer memory that will be used as a reference in further analysis.

Keywords: Modbus TCP, monitoring, Power Meter, LabView

1. PENDAHULUAN

Salah satu energi yang mendapat perhatian khusus adalah energi listrik. Energi listrik adalah merupakan energi yang paling banyak digunakan di Indonesia. Proses pembangkitan energi listrik merupakan salah satu pengguna energi fosil terbesar khususnya minyak bumi, gas alam dan batu bara. Oleh karena itu pemakaian energi listrik yang efisien akan berdampak langsung pada tingkat konsumsi energi fosil.

Gedung adalah merupakan pengguna energi listrik yang cukup besar, di Amerika Serikat energi yang digunakan pada gedung mencapai 40% dari total konsumsi energi di negara tersebut [2]. Gedung yang lebih efisien dalam penggunaan energi listrik langkah penting dalam untuk mengurangi konsumsi energi dan emisi karbon. Salah satu cara yang digunakan untuk mengefisienkan penggunaan energi pada gedung adalah penggunaan sistem **Smart Building** (Gedung Cerdas) diantaranya adalah melakukan monitoring terhadap penggunaan energi. Sistem gedung cerdas dapat menghemat penggunaan energi sebesar 10% sampai 15% [4].

Pada penelitian ini peneliti akan mengimplementasikan sistem bangunan cerdas pada gedung komersial. Sistem akan membaca data masukan berupa tegangan, arus, frekuensi secara *real time*. Juga akan menghitung faktor daya dan konsumsi energi

listrik secara real time yang dihasilkan dari formula turunan yang diolah oleh sistem. Hasil akuisisi data tegangan, arus, frekuensi, daya dan faktor daya untuk disimpan pada memori komputer yang akan dijadikan acuan pada analisa lebih lanjut.

2. TINJAUAN PUSTAKA

2.1 SCADA (Supervisory Control And Data Acquisition)

Seiring dengan meningkatnya jumlah informasi yang ada dan tersebar di berbagai tempat, diperlukan adanya suatu teknologi yang mampu menampilkan informasi-informasi itu, dan kemudian menyajikannya kepada *user* dalam bentuk yang mudah untuk dimengerti, misalnya menampilkan informasi di *web* dalam bentuk grafik. Selanjutnya informasi tersebut juga dapat dijadikan sebagai sebuah laporan yang dapat diolah untuk kepentingan analisis selanjutnya.

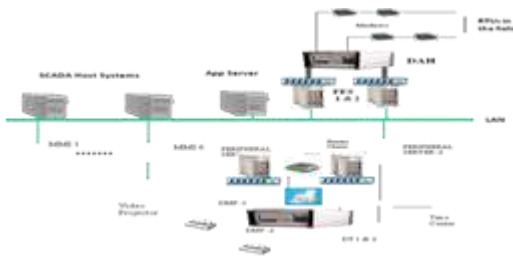
Sebuah jaringan komputer yang terintegrasi dengan peralatan yang dapat membaca sesuatu besaran dari jarak yang jauh adalah bagian dari sistem SCADA. SCADA (*Supervisory Control And Data Acquisition*) didefinisikan oleh dua sumber tipikal:

1. SCADA adalah teknologi yang memungkinkan seorang pemakai untuk mengumpulkan data dari sebuah

fasilitas yang lebih jauh dan/atau mengirimkan instruksi kendali terbatas pada fasilitas itu [1].

2. SCADA adalah suatu sistem yang beroperasi dengan isyarat kode di atas saluran komunikasi agar dapat mengendalikan peralatan RTU (*Remote Terminal Unit*). IEEE Std C37.1-2007 (Revision of *IEEE Standard C37.1-1994*), *Definition, Specification and Analysis of Systems Used for Supervisory Control, Data Acquisition, and Automatic Control* [3].

Arsitektur tipikal sistem SCADA terdiri dari: Human Machine Interface (HMI), Master Terminal Unit (MTU), Remote Terminal Unit (RTU), dan Field Data Element/Plant. Komunikasi dari MTU ke RTU atau sebaliknya dapat menggunakan komunikasi melalui radio communication, wired network, PSTN, intranet/internet, atau wireless network [1].



Gambar 1 Arsitektur SCADA

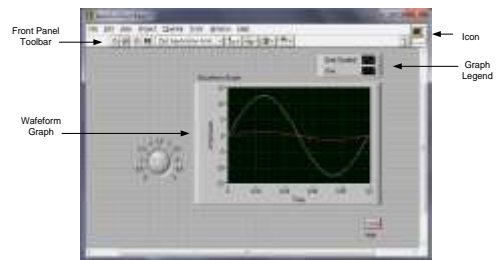
2.2 LabVIEW

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) adalah *software* yang dikembangkan oleh *National Instrument* pada tahun 1986. Labview merupakan *software* khusus yang dikembangkan dan digunakan untuk pemrosesan dan visualisasi data dalam bidang akuisisi data, kendali dan instrument, serta otomatisasi industri [3].

Lingkungan pemrograman LabVIEW memiliki dua bagian utamayaitu *front panel* dan *block diagram*. *Front panel* digunakan untuk berinteraksi dengan pengguna pada saat program dijalankan. pengguna dapat mengontrol program, mengubah input dan memantau data secara *real time*. Sedangkan blok diagram digunakan untuk membuat rangkaian fungsi-fungsi dengan cara memasukkan fungsi-fungsi yang diinginkan

dan menghubungkannya satu dengan lainnya.

Gambar 2 menunjukkan contoh tampilan *front panel* LabVIEW, dimana pada tampilan kotak segi empat dengan bentuk grafik sinyal sinus adalah besaran tegangan yang terukur dari input simulasi dari internal LabVIEW. Selanjutnya tampilan di samping berupa sebuah *knobs* untuk mengatur besar nilai amplitudo dari gelombang



Gambar 2 Front Panel pada LabVIEW

Pada blok diagram dapat terlihat gambar fungsi-fungsi yang dihubungkan satu dengan lainnya seperti diagram *flow* yang di dalamnya terdapat fungsi-fungsi yang dapat mengukur dan menghitung serta fungsi lainnya seperti terlihat pada gambar 3.



Gambar 3 Blok Diagram pada LabView

2.3 Power Meter Schneider Tipe PM 800

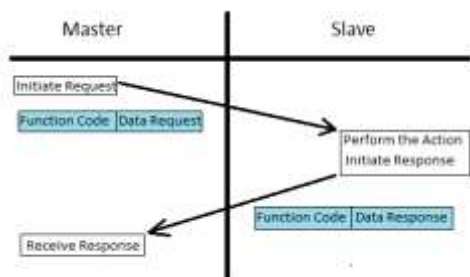
Power Meter adalah sebuah meteran listrik multifungsi yang dapat berfungsi sebagai instrumentasi digital dan perangkat akuisisi data serta pengontrolan daya. Power Meter PM 800 bisa menggantikan berbagai fungsi relay, transduser dan komponen lainnya serta dilengkapi dengan komunikasi RS485 untuk pengintegrasian ke dalam suatu sistem monitoring dan pengontrolan daya. Power Meter ini dapat digunakan di industri, gedung perkantoran, gedung kampus dan gedung komersial lainnya.



Gambar 4 Power Meter Schneider Tipe PM 800 dan aksesoris

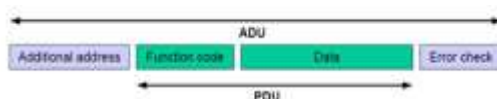
2.4 Pengetahuan Dasar Protokol Modbus

Protokol modbus dibuat oleh perusahaan Modicon tahun 1979 dan sampai sekarang menjadi salah satu protokol komunikasi standar yang dipakai dalam otomatisasi pengelolaan Gedung, Proses Industri dan lain-lain. Jaringan Modbus terdiri dari master dan beberapa slave. Master yang berinisiatif memulai komunikasi antara lain menulis data, membaca data dan mengetahui status slave. Permintaan master disebut juga sebagai *request* atau *query*. Slave hanya bersifat pasif/menunggu atau dengan kata lain slave hanya merespon jika ada permintaan/*query* dari Master [5].



Gambar 5 Transaksi komunikasi Modbus

Pesan yang dipertukarkan antara master dan slave disebut *frame*. Ada dua jenis frame Modbus: *Protocol Data Unit* (PDU) dan *Application Data unit* (ADU). Frame PDU berisi kode fungsi diikuti oleh data. Kode fungsi merupakan tindakan untuk melakukan dan data merupakan informasi yang akan digunakan untuk tindakan ini. Frame ADU menambahkan lebih sedikit kompleksitas dengan bagian alamat tambahan. Frame ADU juga menyediakan beberapa pengecekan *error*. Baik ADU dan frame PDU mengikuti encoding *Big-Endian*.



Gambar 6 Frame Modbus

2.5 Penyimpanan Data pada Modbus

Pada protokol modbus terdapat 4 buah jenis penyimpanan data dengan panjang masing-masing 16 bit.

Primary Tables	Object Type	Type of
Discrete Input	Single bit	Read-Only
Coils	Single bit	Read-Write
Input Registers	16-bit word	Read-Only
Holding Registers	16-bit word	Read-Write

Gambar 7 Penyimpanan Data pada Modbus

1. Diskrit Input
Diskrit input digunakan untuk mengetahui status relay apakah sedang ON atau OFF. Input relay bersifat read only bagi master dan hanya bisa dirubah oleh slave saja. data tsb disimpan di register 10001 sampai 19999
2. Coil
Pada awalnya jenis data ini digunakan untuk mengaktifkan coil relay. nilai jenis data ini ON atau OFF. Coil mempunyai panjang 16 bit, sehingga untuk mengaktifkan ON dengan cara memberi nilai FF00H dan 0000H untuk OFF. Data FF00 dan 00 disimpan di register 00000 sampai 09999
3. Input Register
Input Register digunakan untuk menyimpan data analog dengan range nilai 0 ~ 65535. Input register bersifat *read only* bagi master. data ini disimpan di register ber nomor 30001 sampai 39999
4. Holding Register
Holding register digunakan untuk menyimpan nilai dengan range 0~65535. Register ini mempunyai alamat register 40001 sampai 49999

3. METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Penelitian dilakukan dalam beberapa tahapan yaitu:

1. Mengumpulkan data lapangan yaitu arus beban puncak, tegangan, frekuensi dan faktor daya.
2. Mengukur jarak RTU (*remote*) ke MTU (*server*) sebagai pertimbangan dalam pemilihan jenis wireless yang digunakan
3. Menginstal dan melakukan setting Power Meter PM 810 sebagai RTU

4. Membangun Aplikasi SCADA dengan LabView 9.0 pada server
5. Mendesain dan membangun jaringan komunikasi RTU ke MTU
6. Melakukan implementasi protokol modbus TCP pada MTU
7. Melakukan evaluasi terhadap kinerja sistem.

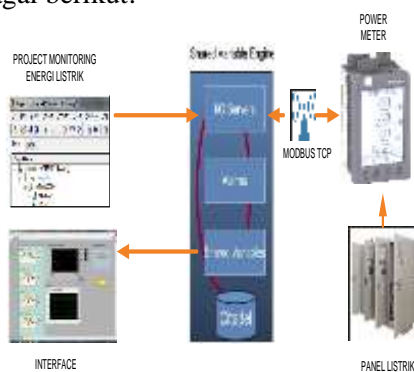
3.2 Peubah yang diamati

Peubah yang diamati pada penelitian adalah:

- a. Tegangan listrik
- b. Frekuensi
- c. Faktor daya
- d. Arus beban puncak

3.3 Rancangan Penelitian

Penelitian dirancang menggunakan sistem jaringan komputer dengan arsitektur sebagai berikut:



Gambar 8 Arsitektur Rancangan Sistem

3.4 Teknik Pengumpulan Data

Teknik pengumpulan data yang dibutuhkan dalam penelitian adalah:

1. Studi literatur
Kegiatan yang dilakukan adalah mengumpulkan data yang berhubungan dengan obyek penelitian melalui buku-buku yang menjadi sumber referensi.
2. Penggunaan sensor alat
Teknik ini menggunakan sensor alat sebagai media komunikasi yang berhubungan dengan server. Sensor mengirimkan data pada peubah yang diamati ke komputer untuk dianalisis.

3.5 Alat dan Bahan Penelitian

Perancangan dan implementasi aplikasi membutuhkan peralatan sebagai

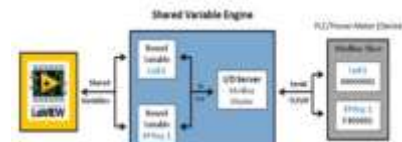
berikut:

1. *Software* Labview.
2. Power Meter Schneider Tipe PM 810
3. Kabel UTP.
4. Komputer Desktop sebagai Server SCADA
5. Protokol modbus TCP
6. Alat Ukur

4. HASIL DAN PEMBAHASAN

4.1 Perancangan Modbus TCP Server I/O dan Shared Variable

Sistem monitoring energi listrik terdiri dari dua bagian utama yaitu pada bagian RTU dan bagian komputer server (MTU). Sistem monitoring ini menggunakan Modbus TCP sebagai protokol komunikasi antara RTU dan MTU. Untuk mengirim data melalui jaringan maka harus menggunakan *Shared Variable* seperti pada gambar berikut [6]



Gambar 9 Shared Variable Engine

LabVIEW Datalog Supervisory Control (DSC) Modul dan Modul LabVIEW Real-Time menggunakan *Shared Variable Engine* untuk mengelola Modbus I/O Server. Modbus I/O Server menangani protokol tingkat rendah yang diperlukan untuk membangun koneksi dan transmisi data antara master dan slave. Modbus Slave I/O server membaca data dari atau menulis data ke perangkat Modbus. Misalnya, Slave Modbus I/O Server pada *National Instruments* kontroler *Real-Time* dan menggunakan *Modbus I/O server* sebagai perangkat *Modbus slave*. *Shared Variable Engine* mengambil data dari Server I/O sehingga data dapat dibaca/tulis oleh LabVIEW [6][7].



Gambar 10 Modbus I/O Server

Setelah membuat proyek Powermeter: powermeter.lvproj maka selanjutnya membangun *Modbus I/O Server*, pada proses ini IP target (PM 810) diinput. Kemudian membuat *Shared variable*, pada penelitian ini menggunakan alamat register PM 810.

4.2 Membangun Aplikasi LabView

4.2.1 Membangun Block Panel

Block Panel adalah tempat pembuatan program. Jendela Block Panel tidak akan terlihat oleh pengguna saat program dijalankan. Pembuatan program dengan cara menempatkan beberapa node dan menghubungkan satu dengan yang lain.

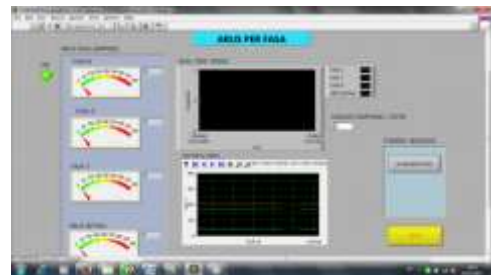
Pada gambar di atas node *Shared Variable* dari arus digabungkan menjadi cluster arus fasa, Cluster arus fasa akan menampilkan nilai arus setiap fasa (R,S,T) dan juga arus netral pada front panel yaitu indikator numerik berupa sebuah meter. Node *Real_time Trend* akan menampilkan nilai dari arus secara *real time*, node tersebut tersambung ke *Chart Graph*. Nilai yang *time* yang ditampilkan di chart graph akan diinput di *real time properties* yaitu pada kolom *Shared Variable List*. Untuk mengatur pengaturan waktu pada sumbu x di *Chart Graph* maka digunakan node *Get Date/Time in Second*, node ini akan mengambil waktu nyata dari sistem dan menampilkannya *Chart Graph*. Menentukan jumlah sampling per detik dilakukan dengan menggunakan node *offset and multiplier* dimana input jumlah sampling per detik akan ditentukan oleh node *wait until next ms multiple* dan inputan jumlah sampling dari *front panel*. Laporan perkembangan nilai arus dari waktu ke waktu akan disimpan di database *Citadel* dan laporannya akan ditampilkan oleh node *NI Hyper Trend*. *NI Hyper Trend* akan berfungsi sebagai *Historical Trend* yaitu menampilkan tren arus dari waktu ke waktu.



Gambar 11 Diagram Block Panel Akuisisi Arus dari power meter

4.2.2 Membangun Front Panel

Front panel merupakan *user interface* dari VI, pada *front panel* kontrol dan indikator dapat ditampilkan. Objek-objek pada *front panel* akan terlihat oleh pengguna saat program dijalankan dan akan secara otomatis memiliki representasi ikonnya pada *block diagram*. Untuk objek-objek yang membawa data yang masuk dari pengguna ke program maupun data yang keluar dari program ke pengguna.



Gambar 12 Front Panel Akuisisi Parameter Arus

Pada gambar 12, *front panel* yang dibangun merupakan representasi dari *block diagram* power meter PM 810, nilai dari *shared Variable* arus akan ditampilkan pada indikator meter secara *real-time*. Meter dari arus akan tampilan dalam suatu cluster, di mana meter akan menampilkan pembacaan nilai arus secara analog dan pada indicator di sampingnya akan ditampilkan digital. Pada tampilan grafik *real time trend* akan ditampilkan nilai parameter arus secara *real time* dalam interval waktu tertentu untuk memudahkan dalam membandingkan nilai dari parameter arus. Jumlah sampling atau cuplikan data per detik dapat ditentukan dengan mengatur nilai pada control jumlah sampling per detik. Rekaman data dapat ditampilkan pada grafik *Historical Trend* merupakan representasi dari *NI Hyper Trend*, *NI Hyper Trend* akan menampilkan data dari database *Citadel*. Pada *historical trend* pengguna dapat menampilkan data arus berdasarkan tanggal yang diinginkan

4.3 Membangun Database Citadel

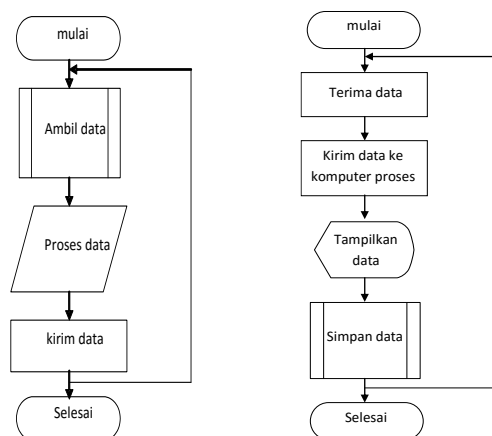
LabVIEW Datalogging and Supervisory Control (DSC) modul menggunakan *National Instruments Citadel database* juga mencakup driver *ODBC Citadel* yang memiliki perintah khusus untuk melakukan

transformasi data, sehingga dapat mengambil, memanipulasi, dan menganalisis data historis secara otomatis dari luar lingkungan LabVIEW.

Database *Citadel* dibangun pada aplikasi *measurement and explorer* dengan menentukan nama dan folder penyimpanan database. *Shared variable* otomatis akan tersimpan dengan menambahkan *project library* ke dalam database setelah mengklik *enable logging* pada *properties shared variable*

PM 810 akan menangani perintah pembacaan data dari sensor kemudian data diproses oleh prosesor power meter. Data akan dikirim menggunakan *Ethernet card* yang terpasang pada PM 8 ECC yaitu *Ethernet card Modbus TCP* yang terhubung ke sebuah *Acces Point*.

Kemudian pada sisi penerima, data akan diterima oleh *shared variable Engine* di mana pada proses ini terdapat Modbus I/O server. Pada aplikasi LabView data akan ditampilkan pada antar muka dan selanjutnya disimpan di database *Citadel*. Untuk menampilkan data dari database *Citadel* digunakan aplikasi *NI Hyper Trend*



Gambar 13 Flowcart pada RTU dan MTU

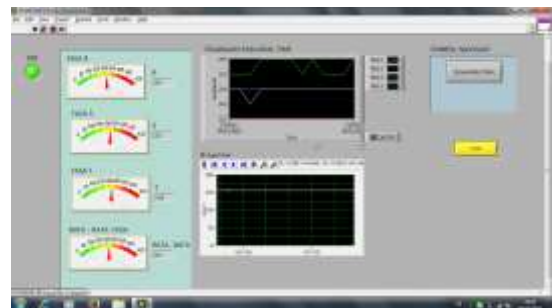
Hasil rancang bangun kemudian diintegrasikan menjadi sebuah sistem monitoring, kemudian dilakukan pengujian dan dikalibrasi. Pada sisi RTU dilakukan terminasi pada panel listrik, pengkoneksian PM 810 dengan PM 8 EE, setting IP dan pengkoneksian ke *Acces Point*.



Gambar 14 Remote Terminal Unit (RTU) dan Master Terminal Unit

Gambar 14 adalah gambar peralatan pada sisi RTU, kemudian dilakukan akuisisi data dari sisi MTU. Hasil akuisisi data setelah dilakukan pembacaan akan dibandingkan dengan hasil pembacaan alat ukur lainnya untuk mengetahui prosentase kesalahan. Proses komunikasi dapat berjalan dengan baik dan data terkirim ke MTU dengan lancar.

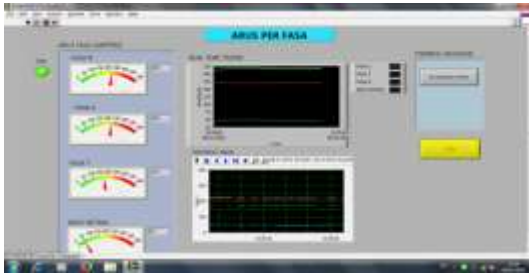
Tampilan antar muka dapat ditampilkan dengan mudah dibaca sehingga dapat membantu pengelola gedung dalam memonitor sistem kelistrikan gedung. Pada penelitian mengakuisisi beberapa parameter listrik dan semuanya dalam berjalan dengan baik



Gambar 15 Akuisisi Tegangan Sistem

Adapaun parameter-parameter tersebut adalah:

- Arus per fasa
- Tegangan per fasa
- Tegangan antar fasa
- Daya Aktif, Daya Reaktif dan Daya Semu total dan tiap fasa
- THD arus dan tegangan total dan tiap fasa
- Energi listrik: kWah, kVAR, kVA total dan tiap fasa
- Faktor daya total dan factor daya tiap asa



Gambar 16 Akuisisi Arus

5. Kesimpulan

Hasil pengujian dan pembahasan disimpulkan bahwa sistem monitoring yang dibangun pada sistem kelistrikan gedung berhasil diimplementasikan untuk memantau parameter energi listrik pada gedung. Sistem monitoring yang dibangun dari modul DSC Labview dapat bekerja dengan baik sebagai sebuah aplikasi HMI SCADA bahkan setara dengan software aplikasi HMI SCADA lainnya. Penelitian ini juga merupakan salah satu penelitian terhadap LabView sebagai sebuah software yang memiliki kemampuan dalam hal *interoperability* yaitu kemampuan berkomunikasi dengan berbagai peralatan yang memiliki protokol yang berbeda khususnya berkomunikasi dengan menggunakan protocol Modbus TCP.

6. Daftar Pustaka

- [1] Bonar Panjaitan., 1999. Teknologi Pengendalian Tenaga Listrik Berbasis SCADA. Indonesia: Prenhallindo
- [2] Han Chen, Paul Chou, Sastry Duri Hui Lei. 2009. The Design and Implementation of a Smart Building Control System, IEEE International Conference on e-Business Engineering
- [3] IEEE Power Engineering Society “IEEE Standard for SCADA and Automation Systems” Approved 5 December 2007
- [4] Yuvraj Argawal, Bharatan Balaji, 2010. Occupancy-Driven Energy Management for Smart Building Automation. Zurich Switzerland: BuildSys
- [5] WWW.Modbus.org
- [6] www.NI.com “Whitepaper”
- [7] www.NI.com “LabView Datalogging and Supervisory Control Module Developer’s Manual