بسم الله الرحمن الرحيم

وزارة التعليم العالي والبحث العلمي
جامعه الشيخ عبدالله البدري

كليه التكنلوجيا

بحث تكميلي لنيل درجة الدبلوم في تقنيه الهندسه الكهربائيه

تخصص شبكات

بعنوان:

تصميم وتنفيذ شبكه تحكم صناعى باستخدام الحاسب الالى
Computer Based Industrial Control System

إعداد :

1.حامد عبدالله حامد
2.عمر عثمان حبوب
3.مصعب عبد الرحمن سند
4.أحمد عوض أحمد
5.خطاب محمد ابراهيم

إشراف الأستاذ:

مهند كمال

مايو2012

الاستهلال

قال تعالي:

بسم الله الرحمن الرحيم

(قل لوكان البحرمدادالكلمات ربي لنفدالبحر
قبل ان تنفدكلمات ربي ولوجئنابمثله مدادا)

صدق الله العظيم

سورة الكهف الايه(109)

الاهداء

إلي معلم البشرية الأول
محمد بن عبد الله (صلي الله عليه وسلم)
الحب الذي ملأالقلوب والشوق الذي غمرالنفوس
إلى...
كل ذرة من تراب هذا الوطن حضر وبادية الذي احتوانا منذ نعومة الأظافر والذي سنبقي عليه أبد الدهر
إلى...
من قال فيهما ربك:
(وبالوالدين إحسانا)
إلي .. من منحونا الحياة حين عصفت بنا رياح الياس وأضاءوا لنا معالم الطريق حين غابت شمس الأمل
آبائنا الأعزاء أمد الله في أيامهم
إلي من رسمت في نفوسناالابتسامة وأرضعن الجمال والادب
أمهاتنا العزيزات أطال الله أعمارهن
إلي ... تلك الشموع النيران علي طريق العلم وهجاً واتقاداً
أساتذتي الأجلاء
إلي ... الشعاع الذي أنار طريقنا والي من واصلوامعنا رحلةالعلم
أخوتنا وزملائنا الأوفياء
الي ... من احتوتنا وإضاءة لنا الطريق بالعلم والمعرفة
جامعة الشيخ عبدالله البدري

شكروعرفان

في البدء الشكر للواحد الاحد و نحمده علي أن مكننا من إخراج بحثنا إلي صورته النهائية ونصلي ونسلم علي المبعوث رحمة للعالمين الذي أخرجنا من الظلمات الي النور. الشكرلأساتذة جامعة الشيخ عبدالله البدري الذين قدموا لنا المساعدة التامة والتوجيه الصحيح في كلما نحتاج إليه

كمانخص بالشكركل من مد يد العون في إخراج بحثنا هذا بالصورة المطلوبة

وجزاهم الله خيرالجزاء

الملخص Abstract

يمكن تقسيم هذا البحث إلي الأجزاء الآتية -:

الحاسب الآلي (السيد)، البرتوكولات ،البرمجيات،الموائم (البوابة) ،الوحدات الطرفية (وحدة الإدخال والإخراج). بحيث تبدأ العملية من الحاسب(السيد)الذي تعمل من خلاله بإرسال إشارة رقمية(ديجتل) فتح أوغلق عن طريق برتوكول أوبرنامج(Profibus) وهذه الاشارة تستقبلها البوابه والتى بدورها تحولها الى برتكول او برنامج modbusلان الوحدات الطرفية (DO-DI) تعمل على modbus.

وحدة الادخال مهمتها جمع البيانات ثم ارسالها الى السيد,و وحدة الاخراج تعمل استقبال البيانات من السيد وارسالها الى الريلى حيث يعمل على تغذيه الكونتاكتر ويكون الخرج الاخير.

Abstract

Our Thesis contains many parts as follow :-

- computer As master , protocols , software , Interface (gateway) , remote terminal units (Input, Output terminal). The process start from computer by sending digital signal serialy to Open or close any device via profibus protocol , the signal which send via master , it will be recived by the gate way.

The gateway work As convertor from profibus protocol to modbus , because the terminal unit useing the modbus protocol .

The function of DI is gathering the feild data and send it back to the master , but for DO is execute the master commands.

الفهـــرس

الاية.....
ii
الاهداء.....**iii**
الشكر والعرفان.....**iv**
الملخص.....**v**
Abstract.....**vi**
1- الفصل الاول: التحكم الصناعي باستخدام الحاسب 1
1-1المقدمه 1
1-1-1مميزات profibus.....1
1-1-2مميزات modbus.....1
1-2أهداف البحث.....2
2-الفصل الثاني :- بنية المنظومه 3
2-1 التركيب الاساسي للمنظومة 3
2-2تهيئة المنظومة 4
2-2-1 برمجيات الحاسب 4
2-2-2كرت الاتصال 4
2-2-3الموائم 5
2-2-4الوحدات الطرفيه.....6
3-الفصل الثالث:- تصميم الوحدات الطرفيه.....7
3-1وحدة الادخال الرقمية.....7
3-1-1تصميم haredware design 7
3-1-2Atmegu128 Mcu.....9
3-1-3تصميم اللوحة المطبوعة 11
3-3-4 1-الاختبار.....12
3-2 وحدة الاخراج 14

3-3 التصميم sehekatic.....14

4- الفصل الرابع soft ware Debagging 17

4-1البرمجيات.....17

4-1-1المبدأ التتابعي.....17

4-1-2تهيئة التابع.....17

4-1-3اختيار اتصال opc.....19

4-2اختبار الاتصال لوحدات الادخال –الاخراج.....20

4-2-1وحدةDI:.....20

4-2-2وحدةDO:.....24

5-الفصل الخامس opc and mcgs communication.....26

5-1تاسيس المشروع على opc.....26

5-2تاسيس المشروع على برمجيه mcgs.....27

6- الفصل السادس: تطبيقات.....28

6-1 تصميم اللوحه الكهربائية.....28

الخاتمة و التوصيات 29

المراجع.....30

الملاحق.....31

الفصل الاول

Computer based industrial control system

التحكم الصناعي باستخدام الحاسب الالى

1-1المقدمة : Introduction

يمكن تصنيف نظم التحكم الصناعي (ICS) للأنظمة الآتية :

1-نظم السيطرة الإشرافية وجمع البيانات SCADA
2-نظم التحكم الموزعه DC3
3-أنظمه التحكم المبرمج PLC

يستخدم نظام ICS في الصناعة ، الكهرباء ، المياه ، البترول.

نظرية العمل لنظام ICS تبدأ بجمع البيانات من الوحدات الطرفية RTU ومن ثم تنفيذ توجيهات الآلية أو توجيهات المشرف لوحدات التحكم ألطرفيه ، كما تصنف وحدات RTU إلي مدخلات / مخرجات I/O، وعادة ما يعبر عنها بالوحدات الحقلية Field device والتي تتصل بدورها مع صمامات ، قواطع ، ريلهات ومحساسات .

يهتم البحث بتصميم منظومة تجمع بين نظام التحكم الموزع DCS في التكوين ونظام SCADA .

نجد أن كل هذه الأنظمة تستخدم نظام الاتصال (الناقل الحقلي fieldbus) وذلك لربط الوحدات الطرفية RTU بوحدة التحكم .

الناقل الحقلي fieldbus:

هو برتوكول لشبكات الحاسب الصناعية
Industrial computer network protocis

يستخدم لأنظمة التحكم ألموزعهDCS ذات الزمن الحقيقي(realtime) حيث توجد عدة برتوكولات قياسية منها :
Interbus ، Foundation ، CAN ،Modbus ، Profibus

يستخدم البحث كل من Modbus-Profibus وذلك للاستفادة من الايجابيات لكل منهم والتخلص من السلبيات .

1-1-1مميزات Profibus :

1- **برتوكول قياسي ، يتوفر من أكثر مصنع .**
2- **تقليل ألكلفه وذلك بتقليل عدد الكوابل المستخدمة .**
3- **سهولة اكتشاف الأعطال بالمنظومة .**
4- **السرعة العالية والتي تتراوح بين 9.6kbps - 12 mbps.**

1-1-2ميزات Modbus :

يتميز بسهولة البرتوكول وبالتالي سهولة برمجة الوحدات ألطرفيه .

2-1أهداف البحث :

استخدام برتوكول profibus للخصائص التي يتميز بها واستخدام برتوكول Modbus لسهولة البرمجة .

يهدف البحث ألي تصميم وحدات طرفيه RTU ومن ثم برمجتها ببرتوكول Modbus حتي تتفاهم مع بوابة الطريق Gate way.

والتي بدورها تم برمجتها من بحث سابق بحيث تترجم برتوكول Profibus الي Modbus .

الفصل الثاني

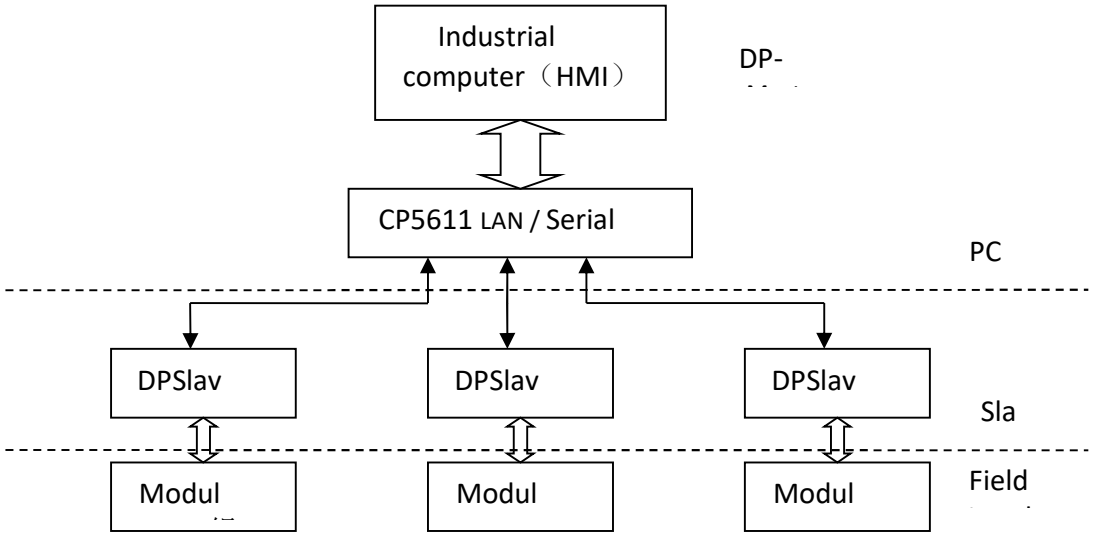
بنية المنظومة System structur

هـذا الفصـل يتنـاول التركيـب العـام للمنظومـة والـذي يبـدأ مـن كـرت الاتصـال cp5611 ومـن ثـم البوابـة Gateway والتـي تحـول مـن برتوكـول profibus إلـي modbus ومـن ثـم الوحـدات الطرفيـة RTU التـي تمثـل فـي (Input-output)او (Analogu-Digital)والتـي تعتمـد فـي تصـميمها علـي المايكرو كنترولـه Atmega 128 .

2-1 التركيب الأسـاسي للمنظـومة Basic system structure

يمكـن ان تمتـد المنظومـة الـي 32 محطـة بمسـافة قصـوي 1200meter بـدون repeater تسـتخدم برتوكـول profibus والـذي يعتمـد علـي وصـله RS485فـي الطبقـة الفيزيائيـة بـدلا مـن Rs232 وذلـك لان نظـام RS485اكثـر ملائمه للبيئة الصناعية من طول مسافه وربط اكثر من slave.

تسـتخدم المنظومـة الحاسـب كسـيد master باسـتخدام كـرت الاتصـال CP5611 الـذي يولـد برتكـول profibus والـذي يـربط المنظومـة بـالتوابع slaves التـي تتمثـل فـي وحـدات الادخـال والاخـراج الرقميـةDIوDO التـي تعمل علي برتوكول Modbus كما موضح بالشكل (1-2) و (1-1)



شكل(2-1)البنية الاساسية

نجـد ان كـل محطـة Station او تـابع Slave يتكـون مـن بوابـة الطريـق gateway والتـي تعمـل علـي توافـق البرتوكـولات والتـي تتصـل

بوصلهprofibus من طرف وتتصل بوحدات الادخال والاخراج DI و DO من طرف اخر بوصله برتوكول Modbus ، كما نجد ان الاتصال يستخدم التوالي Serial ، أي وجود خطين احداهما لارسال البيانات Tx واخر لاستقبال البيانات Rx .

2-2 تهيئة المنظومة configuration

2-2-1برمجيات الحاسب pc software

تحتاج المنظومة الي عدة برمجيات منها لاختبار الاتصال ،تهيئة المنظومة و مراقبة المنظومة .

نسبتا لان البرتوكول profibus هو ملك لشركة سيمنز الالمانية لذا تستخدم برنامج SIMATIC STEP7 V5.42والتي تستخدمة شركةSEIMENS لبرمجةPLC S7-300وتهيئة شبكاته .

تحتاج المنظومة لنظام OLE الصناعي او ما يسمى (OPC) Object Linking Embedding For Process Control لذا نستخدم برمجية Siemens Simatic Net 2006v6.4 اخيرا، تستخدم برمجية MCGS (Monitor and control genereted system) لاغراض التحكم والمراقبه.

خصائص التشغيل لهذه البرمجيات كما يلي :-

1- Siemens step7 software

تستخدم لتهيئة الاتصال من خلال تحديد العنونه للتوابع slave وسرعة الاتصال وذلك من خلال اضافة فايل تعريف التابع او مايعرف بـ GSD لدي شركة سيمنز .

2-SIMATIC NET soft ware

تحتوي علي عدة مركبات ، تستخدم في البحث من اجل استخدام الخادم OPCserver ، يستخدم نظام opc للربط ما بين كرت الاتصال cp5611 وبرمجية المراقبة MCGS ، كما يمكن تعريف المتغيرات الرقمية اذا كانت BIT او word من خلال نظام opc .

3- MCGS software

احدي برمجيات التحكم والمراقبة تتميز بالعرض الديناميكي وعرض خصائص التشغيل للوحدات الطرفية . وهي بمثابة برمجيات ربط المشغل بالماكينة human machine interface (HML)

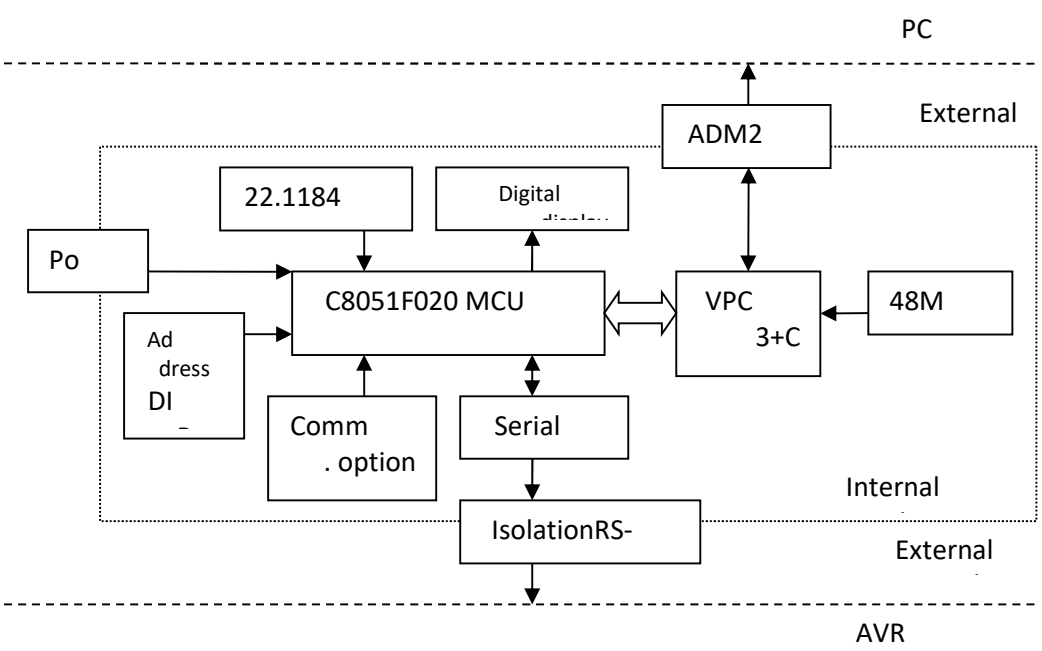
2-2-2 كرت الاتصال CP5611 NET WORK INTERFACE CARD

وهو معالج لعدة اغراض اتصال ، يصنع من قبل شركة سيمنز وهو يدعم برتكول profibus-DP .يتم ادراجة بمجاري الحاسب pcibusslotكما يمكن ان يستخدم كسيد او تابع ، يستخدم في البحث سيد الشبكة master وذلك لقياده التوابع للاتصال Polling Station for communication من خلال تمرير حلقه حديث token passgring والاستجابه response بفتره دورية ثابتة Cycle time الشكل (2-2) يوضح كرت الاتصال CP5611.

شكل (2-2) يوضح كرت الاتصال CP5611

2-2-3 الموائم :profibus – DP Adapter

اهم الاجزاء بالمنظومة هو الموائم الذي يعمل علي ترجمة برتوكول profibus حتي يتلائم مع برتوكول Modbus ، كما يجب ان يكون اكثر استقرارا وله مقاومة عالية ضد التدخلات unti-interference الكهرومغنا طسيه تم تصميمة لاتاحه العنونة من خلال اربعة مفاتيح وكذلك عرضها من خلال عارضة T-Segmentيستخدم الموائم دائرة متكاملة chip مصنعة من قبل شركة سيمنز VPC3 والتي تحتوي على برتوكول profibus ، كما تتبع هذه IC عائلة AS IC ،ولتحقيق عملية الاتصال يجب ان تتصل IC بـ 8bitup عن طريق 8 خيوط (توازي) لهذا الغرض يستخدم الموائم المايكروكنترولة C8051F20والتي تتبع الي عائلة 8o51mcu ومن ثم برمجة هذه المايكروكنترولة باستخدام توجيهات المالك سيمنز لتحقيق هدف الاتصال . وهذه الاخيرة بدورها تتصل توالي serial port باستخدام modbus مع الوحدات الطرفيةDI و DO كما موضح في الشكل (2-3).



الشكل (3-2)بنية الموائم

أخيـــرا يســـتخدم المـــوائم ADM2486 وذلـــك لتحويـــل المنطـــق TTL الـــي RS485 لأغـــراض الاتصـــال لبرتوكـــول profibus وكـــذلك max485 فـــي جانب modbus .

4-2-2الوحدات الطرفية : (DI,DO)

الوحـــدات الطرفيـــة RTU تتمثـــل فـــي وحـــدات الادخـــال والاخـــراج امـــا ان تكون رقمية DI، DO او تماثليهAI،AO.

يتنـــاول البحـــث طـــرق التصـــميم والبرمجـــه فقـــط لوحـــدات الادخـــال /الاخـــراج الرقميه DI،DO.

مـــن اجـــل تحقيـــق كفـــاءة – كلفـــة التصـــميم ثـــم تقســـيم الوحـــدة الـــي جـــزئين (جـــزء علـــوي وســـفلي) ، حيـــث توجـــد MCU فـــي الجـــزء العلـــوي امـــا الجـــزء الســـفلي اما ان يعمل علي DI او DO .

المايكروكنترولـــة المســـتخدمة هـــي Atmega 128 مـــن شـــركة Atmel ، حيث يتم برمجتها وفقا لوظيفة الوحدة اما DI او DO .

الفصل الثالث

تصميم الوحدات الطرفية Remote terminal unit Desige

3-1: وحدة الادخال الرقمية Digital Input (DI)

تعمل علي جمع البيانات الرقمية من اجهزة المجال field device (حساسات ، مفاتيح ، ضواغط ، ...) ثم ارسالها الي وحدة الموائم عن طريق برتوكول modbus حيث يتم ارسالها الي الحاسب بواسطة الموائم باستخدام برتوكول profibus .

يمكن لوحدة الادخال DI ان تجمع بيانات من 32 نقطة (32switch) الي ما يعادل 8 bit او 4bytes .

في حالة لا يوجد نفاذ للبيانات data accessمن قبل القائد فان وحدة DI تعمل علي جمع وتخزين البيانات بالمسجل Digital Capture Register ثم الانتظار لطلب قراءة البيانات من قبل القائد Master (الموائم) باستخدام برتوكول Modbusبسرعه بيانات 9600 bps غير متزامن asynchronous،noparity .

3-1-1التصميم hardware Design

تم تقسيمDI الي مكونات صلبة hardware وبرمجة software المكونات الصلبة يتم تقسيمها الي جزئين ، وحدة المعالجة Cpu تتمثل في الجزء العلوي ، وجزء التوصيل البيني لجمع البيانات الرقمية DI interface Acquisition board وتوجد بالجزء السفلي لوحدة وأخيرا يتم تركيب الجزء العلوي علي الجزء السفلي .

تقسيم الوحدة الي جزئين يساهم في مرونه الصيانة ، حيث ان الجزء العلوي Cpu متماثل تماما لوحدة DO بينما الاختلاف يكون برمجيا.

الادوات الصناعية مثل المفاتيح والحساسات Switches and senors يتم توصيلها للجزء السفلي لوحدة DI باستخدام مستوي جهد قدرة 24v ، ومن ثم تحويلة الي المستوي فولتية 5v وذلك للتوصيل البيني مع وحدة Cpu والتي تتمثل في المايكروكنترولة Atmega 128 Mcu لمعالجة البيانات الرقمية والاتصال عن طريق Rs485مع host,تم تصميم المخططات الالكترونية Schematic diagram باستخدام برمجية Protel Dxp 2004 Software.

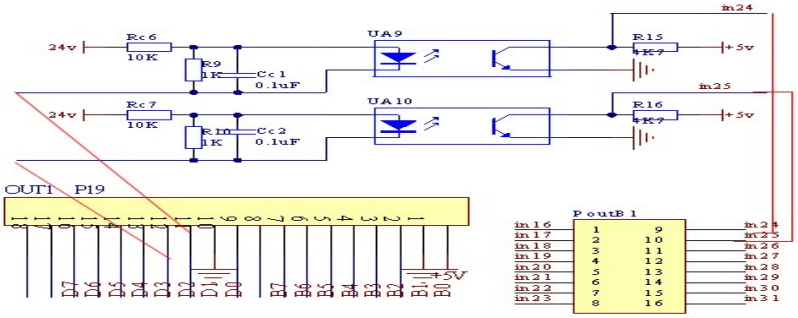
التركيب وطرق التوصيل للوحدة كما موضح ادناه :-

توصل المداخل (32input) الي القارنات الضوئية optocoupler كما موضح بالشكل (3-1) عن طريق header ، وظيفة القارنات الضوئية هي العزل الكهربي كما يوجد Low pass filter مرشح الترددات المنخفضة للحد من الحالات العابرة للمفاتيح او الحساسات.

وكما موضح بالشكل يلاحظ ان عندما توصل المداخل D2,D3 علي سبيل المثال للـ 24GND يسبب في توصيل القارن الضوئي بجانب 24v وهذا بدوره يعمل علي توصيل الارجلIn24،In25 pins للGND5 وهذا الاخير بدوره يعمل علي توصيل مستوي الفولتية المنخفضة 5v للمايكروكنترولة ، كما يسبب في اضاءة LED المقترن بذلك الرجل .

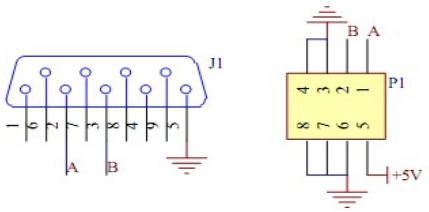
مشابها ، عندما توصل D3 وD2 الي 24v+ ، هذا يسبب في حالة Off للباعث الضوئي للمقارن الضوئي في جانب 24v وبالتالي فان النقاط in24 in25 , تكون في هذه الحالة منطق مرتفع (high-level) 5v .

كمـــا نجـــد بمـــدخل القـــارن الضـــوئي وجـــود مقاومـــة 10k للحـــد فـــي تيـــار الباعـــث الضـــوئي للمقـــارن (للحفـــاظ ضـــمن المواصـــفات المحـــددة وهـــي 1.5-2vوتيـــار 3-5ma) ، كـــذلك وجـــود مرشـــح(o.1mf,1 RCfilter) لعمليـــات الفـــتح الغيـــر متوقعه.

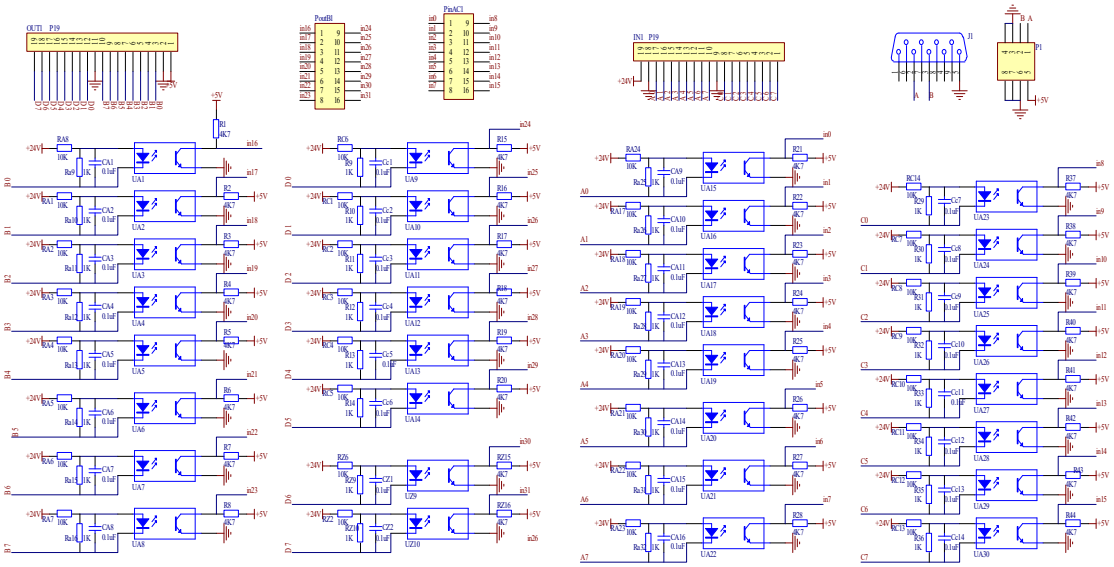


شكل (3-1) Digital Input ACquisition Board

شكل (3-2) يوضح منفذ الاتصال لوحدة D1 والتي تحتوي Rs485connection وذلك لتوصيل الناقل BUS بال P1 header للاتصال مع MCU (الجزء العلوي من الوحدة) .



شكل (3-2) Rs48s connection diagroum



شكل (3-3) المخطط الكلي للجزء السفلي للوحدة D1

The acquisition boards overall circuit diagram

المايكروكنترولة atmega128 تعمل علي معالجة البيانات التي تم جمعها ثم ارسالها الي الموائم بواسطة RTU-modbus ثم بواسطةprofibus الي الحاسب .

2- Atmegu128MCU:

تعمل علي معالجة البيانات التي تم جمعها ثم ارسالها الي الموائم باستخدام برتوكول modbus-RTU .

3- كما نجد ان برتوكول modbus-RTU يستخدم Rs485 في الطبقة الفيزيائية كذلك الامر لبرتوكول profibus ، لذا يجب تحويل مستوي الفولتية للاشارة من منطق TTL الي منطق Rs485 باستخدام Ic MAX485 وهي عبارة Ic يوجد بها مرسل ومستقبل ذات قدرة منخفضة ((Low-power transceivers, drive and receiver) .

يمكن توصيل 128devices في نظام الاتصال Rs485 باستخدام كيبل UTP لتكون منظومة Half-duplex control net work

(حيث ان لا يمكن ارسال واستقبال البيانات بنفس الزمن) .

Maa485 :

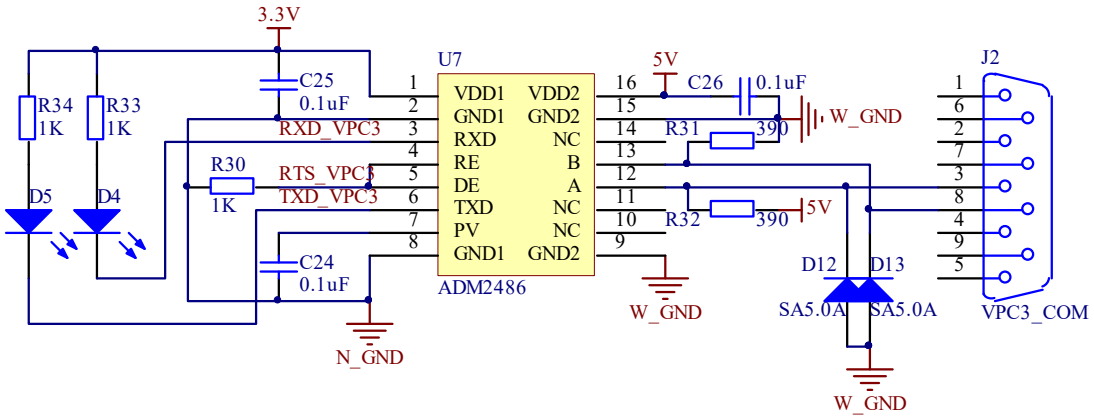
الرجل رقم 1 Pin1 (Ro) هي خرج المستقبل receiver output

pin4 (DI) هي Driver input توصل بالاطراف Rxd ، TXD للمايكروكنترولة.

DE هي تفعيل لـ Driver output بمنطق(high) ، عند (low) تكون ذات مقاومة عاليةhigh impedance A,B هي اطراف Rs485 حيث يتم تمثيل البيانات كالاتي :

- عندما تكون مستوي الفولتية لـ A اعلي من B تمثل المنطق one .
- وعندما A تكون اقل من B فهذا يمثل المنطق Zero .

المقاومات الطرفية Termination resistor تستخدم لمنع انعكاس الاشارة في الناقل Singal reflection on the bus.



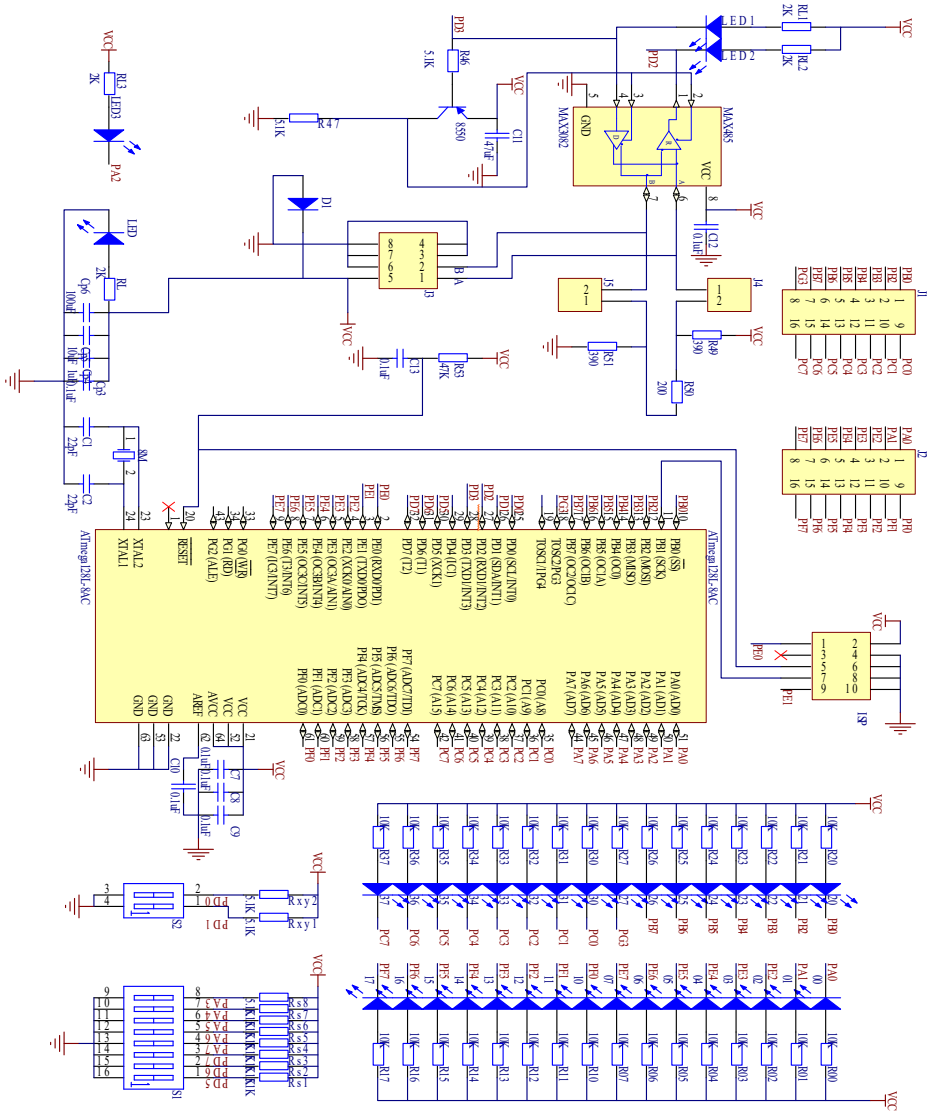
شكل (3-4) schematic diegram of max 485 chip connection

كما موضح بالشكل اعلاه فان الترانزستور 8550 pnpيستخدم للتحكم في تفعيل RS485IC بمعني عندماتكون وصلة الجامع في اعلي قيمه (high) فان IC ترسل بيانات وعندما تكون الوصله منخفضه (low) فانها في حالة استقبال البيانات.

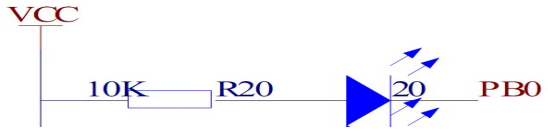
توصـــــــل قاعـــــــدة الترانزســـــــتور لـــــــلmcu عنـــــــد الطـــــــرف txdpin(transmitting) ,عنـــدما يكـــون الطـــرفTXD عـــالي فـــان الجـــامع في حالة ممثاله وبالتالي ارسال البيانات والعكس يمثل استقبال البيانات.

الوصـــله J3 : تعمـــل علـــي توصـــيل mcu بلوحـــة جمـــع البيانـــات الرقميـــه , دايود منظم الجهد DI يعمل علي حماية mcu من الفولتيه العاليه .

الشكل(3-5)يبين الجزء العلوي لوحدة DI والتي تحتوي علي [Atmega128mcu]cpu مرتبطة بمبينات الحالة indictorstatusLED كما مبين بالشكل (3-6) لاحدي المبينات او مؤشرات الحاله ,فعندما يكون الطرف PBO في حالة LOW,فان LED يصبح في حالة ONاواضاءهLight up,تم توصيل الدايودات بشكل الانود الجامعcommon anode ثم اختيار المقاومه لزيادة عمر LED(بفولتية1.5-2Vوتيار3-5ma) .



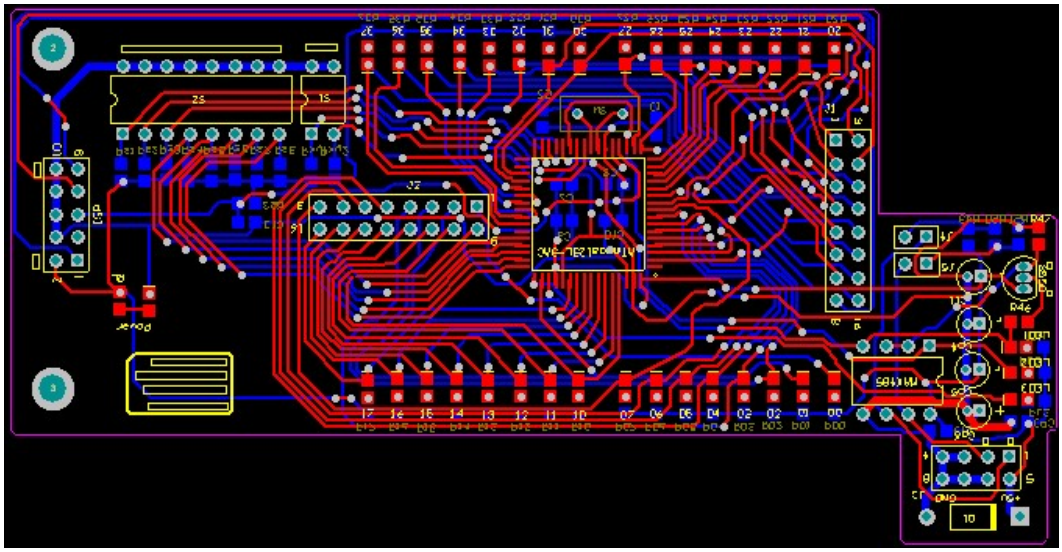
the upper part of DI modele diagneur(3-5) شكل



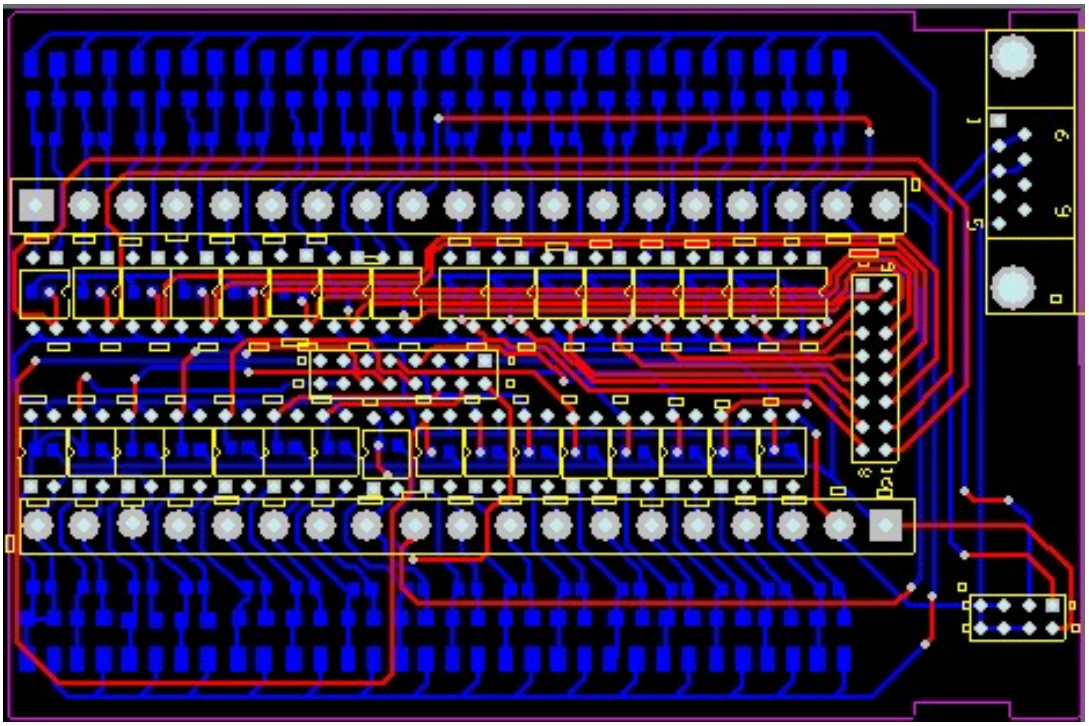
شكل (3-6) port status scheratic circuit

3-1-2تصميم اللوحة المطبوعهPCB Design

تبعا لمتطلبات التركيب نجد ان الجزء العلوي يأخذ شكلا كما موضح بالشكل ادناه ليكون مناسبا لتركيبه علي الجزء السفلي وكذلك ليفي بابعاد الصندوق.



شكل (3-7) PCB upper plate



شكل (3-8) the lower board PCB

التصـــــميم المكتمـــــل لوحـــــدة DIكمـــــا موضـــــح ادناه،حيـــــث تـــــم تركيـــــب الجزءالعلوي علي السفلي.

3-1-3الاختبار hard ware test

1- يجري الاختبار الاولى لقصر الدائرهshort circet عند اطراف مصدر القدره5v,24v في اللوحـــــه الســـــفليه.وكـــــذلك يجـــــري اختبـــــار التوصـــــيله باســـــتخدامDMMللجـــــزء الســـــفلي والعلـــــوي ثـــــم التوصـــــيل بينهم،كـــــذلك اجـــــري اختبار للLEDفي الجزء العلوي.

2- تنزيل البرنامج program downloading

يتم كتابة البرنامج وتنفيذه للـAVRmcuباستخدام برمجيةICC platform .

كما موضح بالخطوات الاتيه-:

***عند دخول القائمة الرئيسيه لبرمجية**

Image craft development (ICC7) for AVR

-click on project→new, in the next pop-up window saves

The project into specific folder as show in fig (3-10)

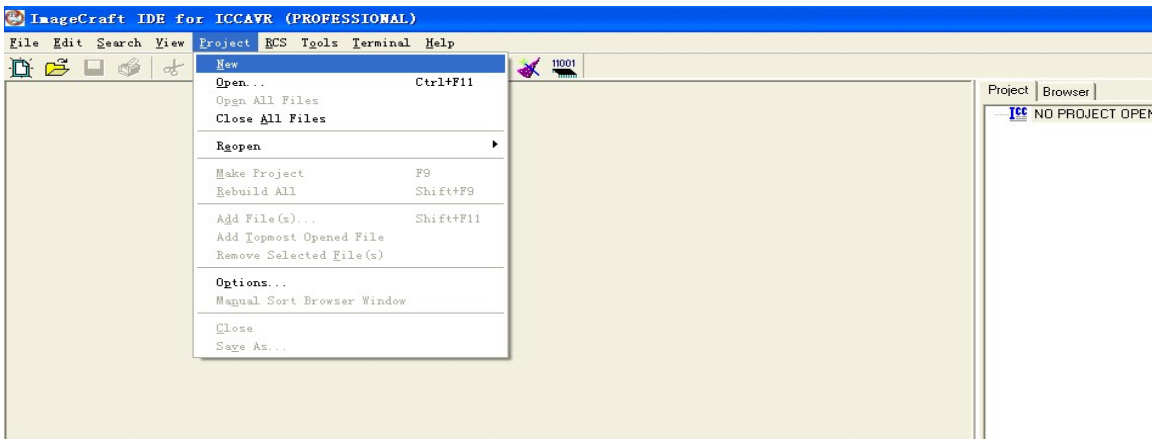
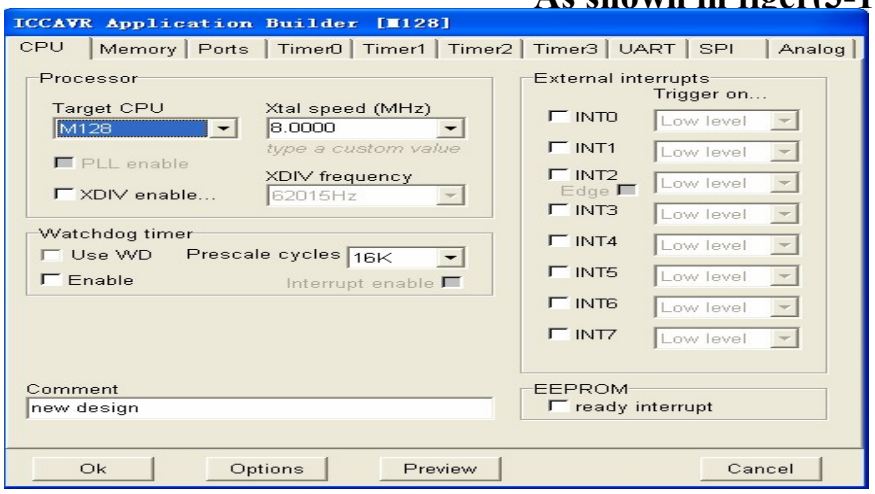


Figure (3-10) new projects in the ICC

-click tools →application builder, select m128

For trgetmcu , then set the xtal speed to 8mhz

As shown in figer(3-11)



CPU initialization settings window(3-11) شكل

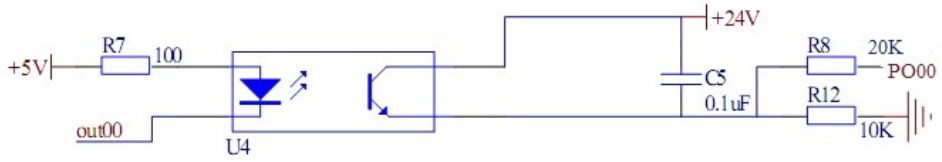
3-2وحدة الاخراج DO module

الوحدة الطرفيهDO تستقبل البيانات الرقميه من الموائم للتحكم في ادوات المجال Field device (للتحكم في الادوات الرقميه الخارجيهexternal digital out put حيث يكون لها حالتين فقط (on or off) وقدتتمثل هذه الادوات في ريلهاتrelay ،صماماتsolenoid،بيانindicator.... تتكون وحدة DOمن32مخرج اوbyte4بمعدلbit8للبايت ،تستخدم برتوكول modbusRTUللاتصال مع الموائم بسرعة 9600bps ونظام اختبار partiy، ونظام اتصال غير متزامن asynchronous.

3-2-1التصميم

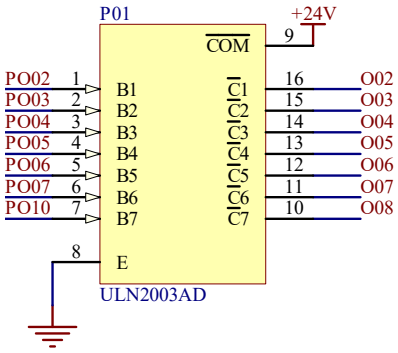
كما بوحدة DIفان وحدة DOتتكون من جزئين احدهما علوي مماثل لوحدةDIلتحسين استخدام اللوحة بينما تختلف فقط في البرامج الموجوده بالAtmega128mcu.

مبدا التتصميم للجزء السفلي يبدا بدائرة العزل القارنات الضوئيه (optocoupler)كما موضح بالشكل(3-12)،استخدام القارنه الضوئيه ياتي من مبدا ان مستوي الفولتيه لمنافذ المايكرو كنترولهe5وهي غير ملائمه للادوات الصناعيه ،لذا يتم مقابلتها بمستوى فولتيه 24vفي الجانب الاخر للقارن الضوئي .



شكل (3-12) isolated input circuit acquisition

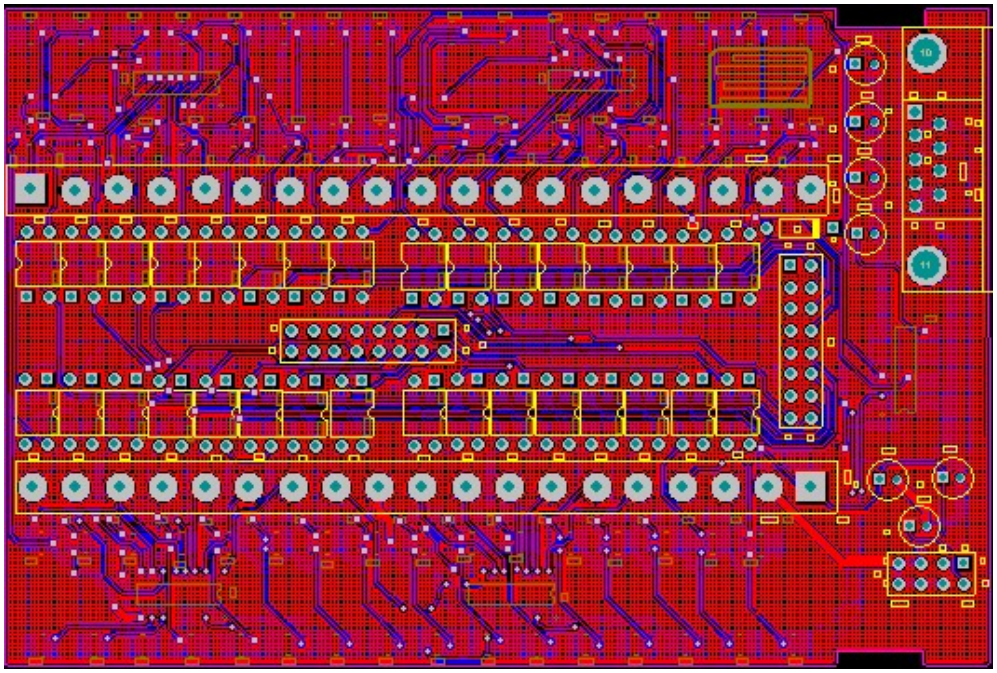
العامل الاخر الذي يجب اتخاذه عند التصميم هو ان السعه الامبيريه لمنافذ mcuايضا غير ملائمه للادوات الصناعيه وبالتالي يجب استخدام driverوهوULN2003كما موضح بالشكل(3-13)حيث يتم تكبير التيار الي 500mA.



شكل (3-13) ULN2003 IC current driver

3-2-2تصميم اللوحة المطبوعة pcb design

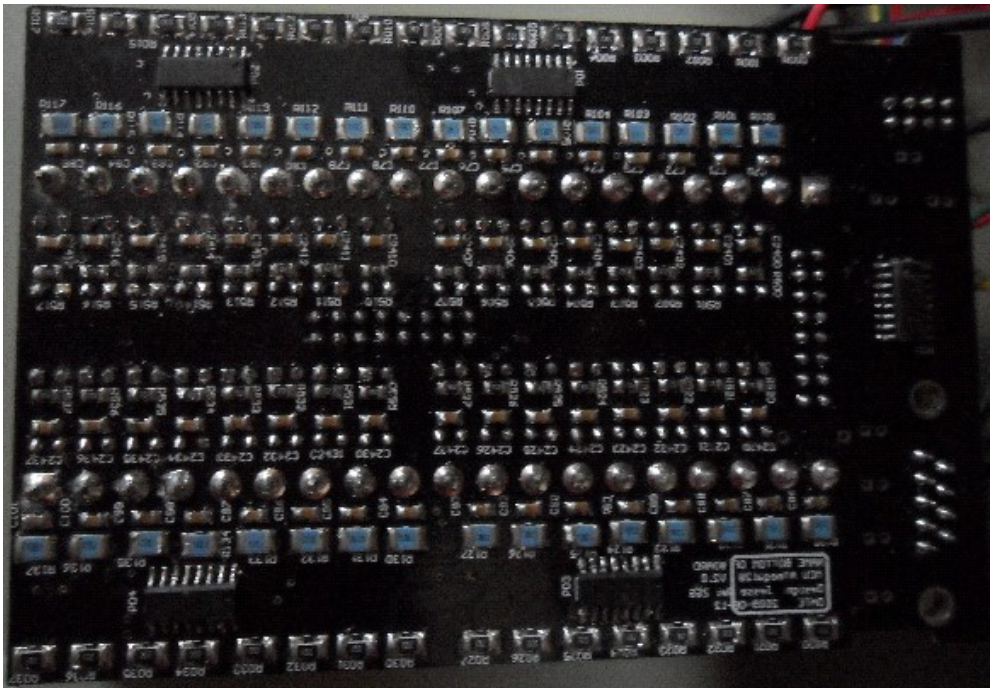
كما ذكرفان الجزء العلوي لوحدةDOيماثل تماما لوحةDI .اما تصميم الجزء السفلي كما موضح بالشكل (3-15)



شكل (3-15)PCB bottom board

حيث يتم تحويل اشارة الخرج الي مستوي فولتيه 24vباستخدام ULN2003ICلتفي بمتطلبات الدوائر الصناعيه.

والشكل (3-16)يوضح الجزء السفلي بعد عمليات تركيب العناصر واجراء اللحام solderingحيث تتم اجراءات خاصه لتثبيتULN2003ICوان لا تزيد فترة اللحام عن اكثر من 5ثواني لتفادي تلفIC.



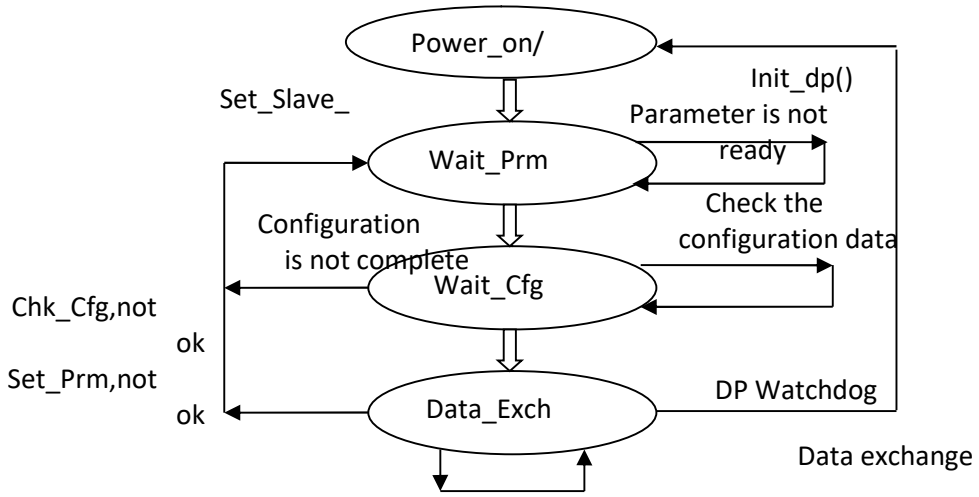
شکل (3-16) DO module finished board

الفصل الرابع
تصميم البرمجيات Software Debugging

4-1البرمجيات

4-1-1 profibus-DP slave state mashine

المبـــدا التتـــابعي لحالـــة الماكينـــة لبرتوكـــول profibus-DP يســـاعد علـــي فهم تتابع Firmware.



شكل (4-1) حالة الماكينه profibus-DP slave state mashine

4-1-2 تهيئة التابع profibus-DP slave configuration

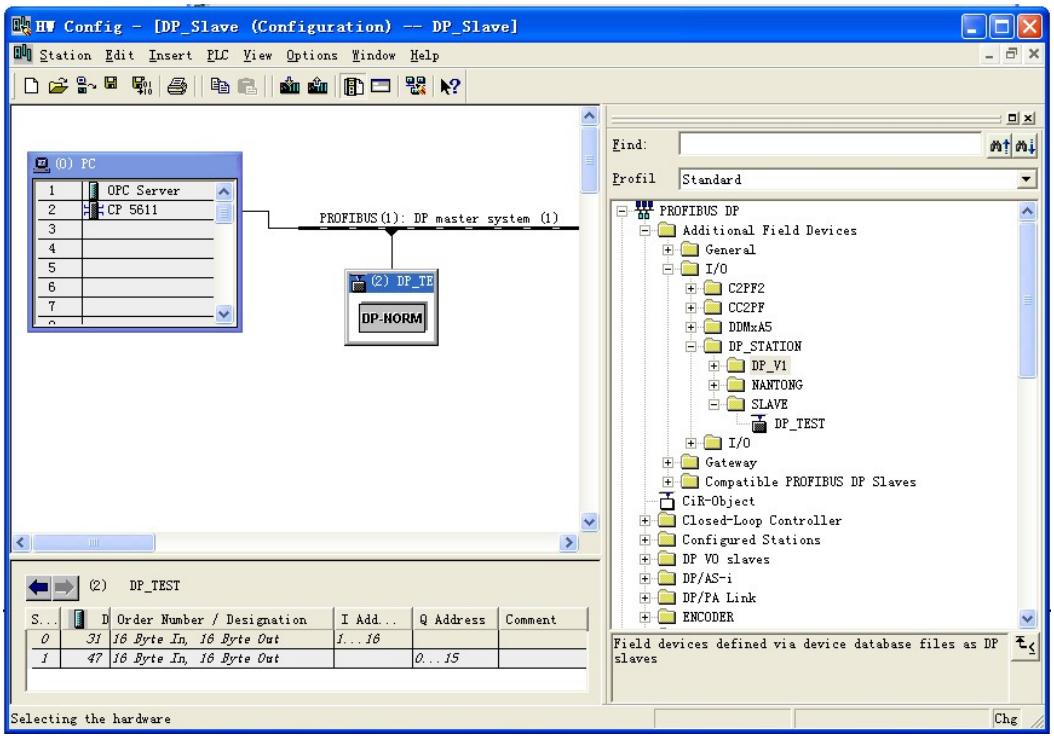
تستخدم برمجية simatic setp-7 لتهيئة شبكةprofibus.

- حيـــث يـــتم اولا تثبيـــت الفايـــل الـــذي يوصـــف التـــابع GSD علـــي البرمجيـــه حتي يتمكن السيد master من الاتصال بشكل صحيح.

- كما ذكر في فصول سابقه فان الشبكه تستخدم نظام opc للربط البيني بين التابع والبرمجيات device programming inter face ومن ثم استخدام cp5611 card لتوليد البرتوكول بعنوان4.

- يـــــتم ضـــــبط cp5611 مـــــن خـــــلال برمجيـــــة step-7 بســـــرعة 1.5mbpsونوعيهprofibus-DP ويعمل كسيد master .

يـــتم اضـــافه التـــابع بعنـــوان 2علـــى شـــبكه profibusكمـــا بالشـــكل(4-2)اخيـــرا يتم compling وتنزيل downloading.



project confguration is complete(4-2) شكل

4-1-3 اختبار اتصال Opc communication test

نجد ان برمجية SIMATIC NET توفر ادوات OPC حيث يستخدمOPC لنقل البيانات اذا كانت DIأو DO ،حيث يتم انشاء مجموعه تحت عائلة opc simatic net.DP ومن ثم اختيار التابع ثم تحديد نوعية البيانات DOاوDI كما موضح بالجدول (2-4)

Module	Power	External power	Signal	Function code	Address	Communication data
DI	5V	24V	Input High (24V) low (0V) level	02	1	4 0-FF bytes
DO	5V	24V	Output high (24V) low (0V) level	01,05	2	4 0-FF bytes
AI	15V	15V	Input 4-20mA current	04	3	Integer 0-1023
AO	24V	24V	Output 4-20mA current	03,06	4	Integer 0-255

الجدول(2-4) مواصفات الوحدات الطرفية

4-2اختبار الاتصال لوحدات الادخال - الاخراج

AVR Input Output device communication debugying

كما سبق ذكره فان الاتصال بين موائم profibus-DP ووحدات الادخال والاخراج يستخدم برتوكول modbus.

نجد ان برتوكول modbus يستخدم 6 دوال اوشفره وهي 01,02,03,04,05,06 [تستخدم في البحث01,02 والتي تدل علي DI,DO علي الترتيب] .مواصفات القدره ،معاملات التشغيل موضحه بالجدول(4-1)

Table (4-1)Device group parameters

كما موضح بالجدول فان كل وحده تحتاج الي مصدر قدره مزدوج5v للmcu و24vللتوصيل البيني بالاطراف pherifial interface . بيانات المجال field data يتم تجميعها باشاره تعتمد علي نوع الوحده وعرض البيانات للوحدات الرقميه32bit، ويستخدم RS485 لطبقة الاتصال الفيزيائيه.

برتوكول modbus له نظامين اما ASC اوRTU ،نجد ان البحث يستخدم نظامRTUمع اختبار خطأ CRC ,ثم اختبار RTU نسبة لإرساله بيانات اكثر من ASC بالإضافه الي انه يملك إطار بيانات بسيط.

4-2-1 وحدة DI:-

1- يستخدم الشفره 02للتوضيح علي انه DI،تتابع التوجيهات كما يلي :

[device address][order no.1][high8-bit of starting register address] [Low8-bit] [Read the high 8-bit register number] [Lower 8-bit] [Lower 8-bit of CRC check sum] [high 8-bit of CRCcheck]
Example:[01][02][00][00][00][20][crclow][crchigh]

1- عنـــوان الوحـــده:- يتـــيح RS485 لعـــدة وحـــدات لتقاســـم البـــاص ,بالتـــالي فان عنونة الوحده مهمه وذلك للانتقال للوحده المطلوبه . كما بالمثال هو1.

2-الشفره02: وهي شفرة الداله وتعني read digital input.

3-عنـــوان الســـجل : وهـــو عنـــوان الســـجل المطلـــوب قـــراءة بياناتـــه الرقميـــه كما بالمثال 0.

4- رقـــم الســـجل : وتمثـــل عـــدد البيانـــات الرقميـــه (مفـــاتيح) المطلـــوب قراءتهـــا من السجلات الرقميه وعرضه 32bitكما بالمثال .

5- اختبـــار الخطـــأ CRC: وعرضـــها 16bit وتمثـــل بيانـــات اختبـــار البيانـــات المرسله.

نجد ان استجابة الوحده هي :

Device response:[Address][order no.01][return the number of bytes][data1][2]……[CRC low][CRC]

Example:[02][04][0C][B1][C2][D2][crclow][crchigh]

1-عنوان اللوحه وهو01

2-عدد البايت المرسل: وهو البيانات n….,2 ,1ويمثل 4byteللمثال.

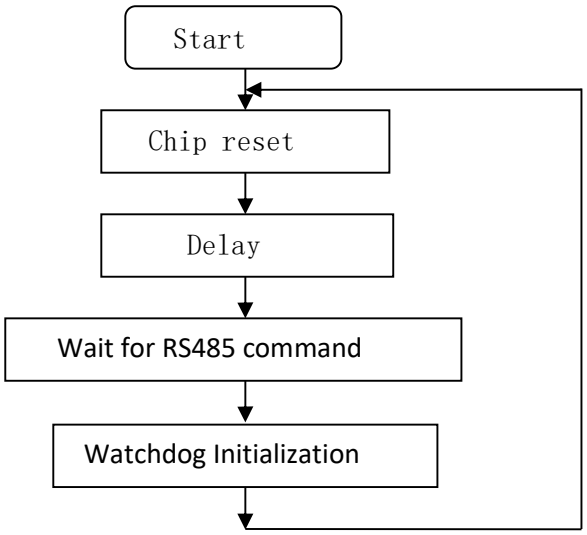
3-n…..Data1:بيانـــات الســـجل، وبمـــا ان كـــل وحـــده بيانـــات عرضـــها 8-bitوبالتالي فهي تمثل حالة 8-switch.

4-Chek sum reduanty check: CRC check.

2- تصميم البرنامج program design:

بعد توصيل الطاقه لوحدةDIيجب علي البرنامج توجيه atmega128chip.

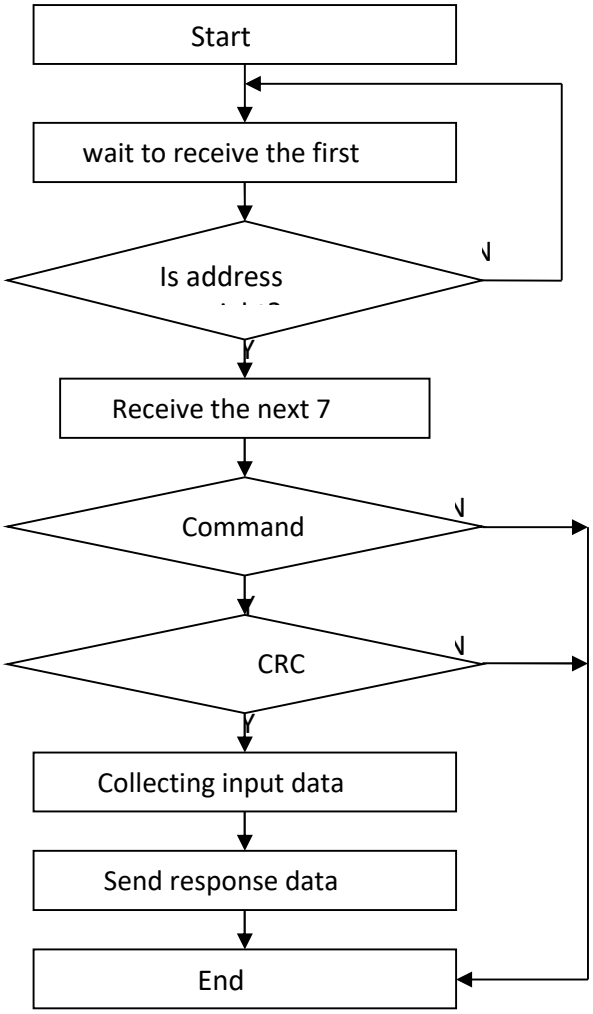
لعملية البدء initialize،وبعد فتره زمنيه قصيره short-delayتدخل مسار لانهائي وانتظار توجيهات pc،كما موضح بمخطط التدفق (4-3)



fig(4-3) DImain program flow charts

وعند استقبال البيانات ، اولا يتم اختبار العنوان للوحده،ثم استقبال 7byteالمتبقيه ، اما اذا كان غير ذلك فيتم رفض البيانات والرجوع الي حالة الانتظار.

بعد استقبال 8byt يتم اختيار شفره الداله 02ثم اختيار CRC كما موضح بمخطط التدفق (4-4)



شکل (4-4) data transceiver module

4-2-2 وحدة DO:

1- شفرة الداله الوظيفيه هي 01,05 read ,writer

01 لقراءة البيانات من DO اما 05 هي لكتابة البيانات علي وحدة DO.

البنيه formatلوحدة DO كالاتي:-

Computer sends:[Address][order No.5][Coil address high 8bit] [Low 8bits][field data high 8bit][low 8bit][CRC Low8bit][high]
Example:[02][05][00][00][ff][00][CRClow][CRChigh]

1-العنونه.

2-شفره تدل علي الكتابة 05.

3-عنوان السجل المراد الكتابه عليه .

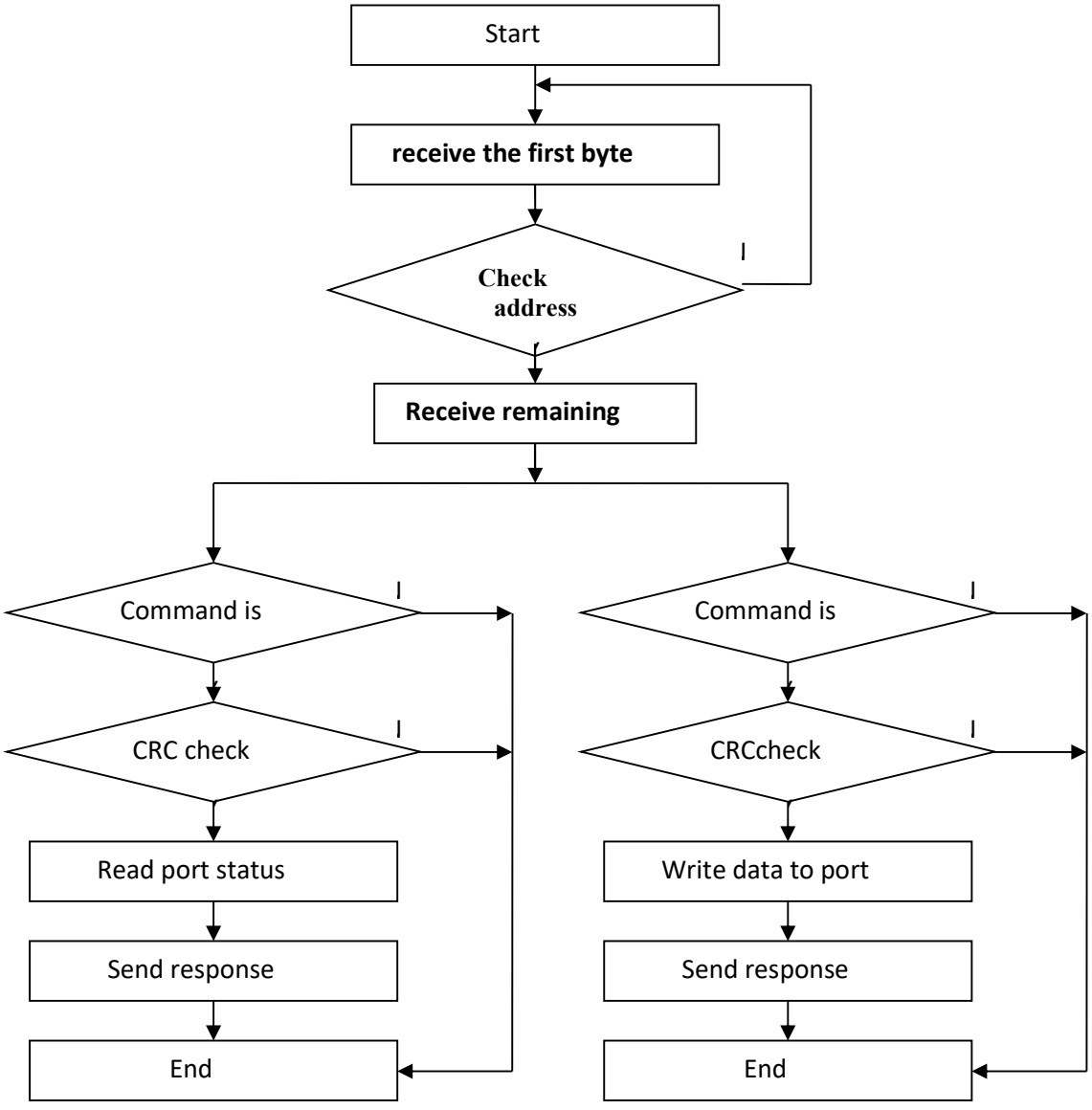
4-البيانات التي سوف يتم كتابتها علي وحدة DO[zero OR one]

كما بالمثال [00][ff]

2-تصميم البرنامج:-

نجد ان إجراءات البرنامج الرئيسي لوحدة DOيماثل وحدة DI،الاختلاف في البدء حيث يتم ضبط النقاط كمخارج .نجد ان التوجيهات 01،05يتم تنفيذ كلاهما لإجراء الفعل المناسب . عند استقبال العنوان الصحيح يتم استقبال البيانات لمتبقيه 7bytes ثم اختبارCRC تبعا للتوجيه 01أو 05 يتم تنفيذ المطلوب.

فإذا كان 01يتم قراءة حالة نقاط DO ويتم إرسالها للـ PC أما إذا كان 05، فسوف يتم كتابة البيانات المرسلة في PC لسجل نقاط الخرج .مخطط التدفق موضح بالشكل (5-4).



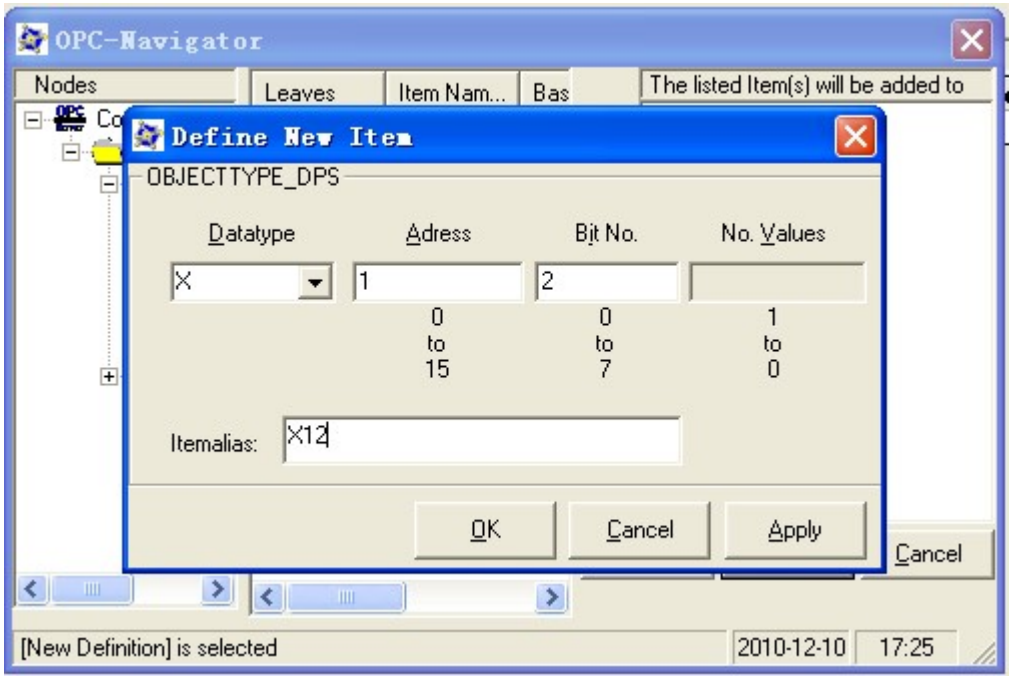
DOdata transceirer flow chart(4-5) شكل

الفصل الخامس
Opc and mcgs communication

نجد ان برمجيات التهيئه المطلوب تثبيتها علي الكمبيوتر الصناعي هي opc,mcgsللاتصال مع منظومة profibus-DPكالآتي:-

5-1 تأسيس المشروع على opc

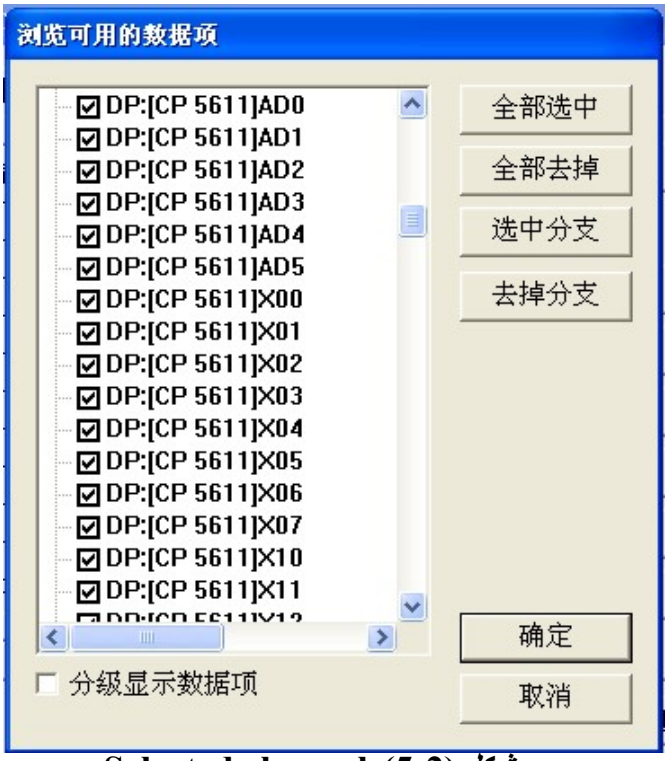
كما زكر سابقا يستخدم opcللربط بين hardwareوبرمجية mcgsحيث يتم تعريف البيانات الرقميه وعنونتها ،علي سبيل المثال يتم اختيار العنصر x12(هذا يمثل البايت الاول من مصفوفة الادخال و2تمثل البت رقم2اي اختيار النقطه رقم10في وحدةDO,DI)كما بالشكل (5-1)



شكل (5-1)definition of channe

5-2 تاسيس الاتصال علي برمجية mcgs

برمجية mcgsتتيح الاتصال معopc(اي انه مدمج في البرمجيه) فيتم اختيار opc simatic net وتبعا لذلك يتم عرض البيانات التي تم تعريفها علي هذا السيرفر(x12) كما موضح بالشكل (5-2)وبالتالي يمكن ربط العنصر x12 مع indicatoفي حالة DI اومفتاح في حالة DO وبالتالي يمكن مراقبة النقطه 10والتحكم فيها ديناميكيا كما موضح بالشكل(5-3).



شكل (5-2)Selected channels

الفصل السادس

تطبيقاتapplication

6-1 تصميم اللوحة الكهربائيه panel design

تم تركيب لوحة كهربائيه تتحكم بموتور حيث يمكن التحكم يدويا بوضع مفاتيح التشغيل علي وضع manualوبالتالى يمكن التحكم بمفاتيح start،stopالموجوده باللوحه أوالتحكم اليا (باستخدام الحاسب) وذلك بوضع المفتاح علي وضع auto حيث يمكن التحكم في تشغيل وايقاف المحرك من الحاسب وكذلك يمكن مراقبة حالات التشغيل للمحرك وتحديد الاعطال باستخدام برمجية mcgs.

في حالة الوضع auto،عند ارسال امر من الحاسب الي وحدة profibus-DPوالتي بدورها ترسل هذه البيانات الي وحدة DO والتي تعمل علي ترجمة البيانات الي اشاره كهربيه تعمل علي اكمال دائرة القارن الضوئي وعن طريق مكبر التيار ULN2003فتتولد اشارة 24vعلي الاطراف والتي تعمل علي التقاط المرحل Relay والذي يكمل دائرة ملف الكنتاكتر coil وبالتالي يعمل المحرك والعكس في حالة الايقاف .

كما يمكن مراقبة حالة تشغيل المحرك (on\off,over load)عن طريق وحدة DI،حيث في حالة حدوث over loadتكتمل دائرة المرحل Relay الذي بدوره يغذي 24v لاطراف DI والتي سوف تترجم الي اشارة5v باستخدام القارن الضوئي لوحدة mcuوالتي سوف ترسل هذه البيانات تسلسليا الي وحدة profibus-DP والتي بدورها الاخيره ترسلهم الي الحاسب.

الخاتمة والتوصيات
Concolution and outlooks

تم تصميم الأجزاء الصلبة الوحدات الطرفية RTU والمتمثلة في DIو DO ، كما تمت برمجتها لبرتوكول modbus.

تم ربط المنظومة وأوضعت تحت الأختبار باستخدام البرمجية MCGS واتضح لنا الأتى :-

- يمكن التحكم والمراقبة لل32 أداة (كنتاكتر مثلا) بتأخير زمنى 2 ثانية.
- تتميز المنظومة بدرجة معقولة من الاستقرارية ، حيث وضعت تحت الاختبار لأكثر من يوم كامل فى حالة تشغيل .
- تعمل المنظومة بشكل آلى عند انقطاع التيار الكهربائى ، اى لا تحتاج الى أعادة ضبط Reset .
- استجابه وحدة الاخراج DO أعلى من استجابة وحدة الادخال DI .

التوجيهات :-

1- البرنامج يحتاج الى اعادة ضبط زمنى و كذلك تاخير حتى تكون استجابه وحدات الادخال / الاخراج عالية و مستقرة .

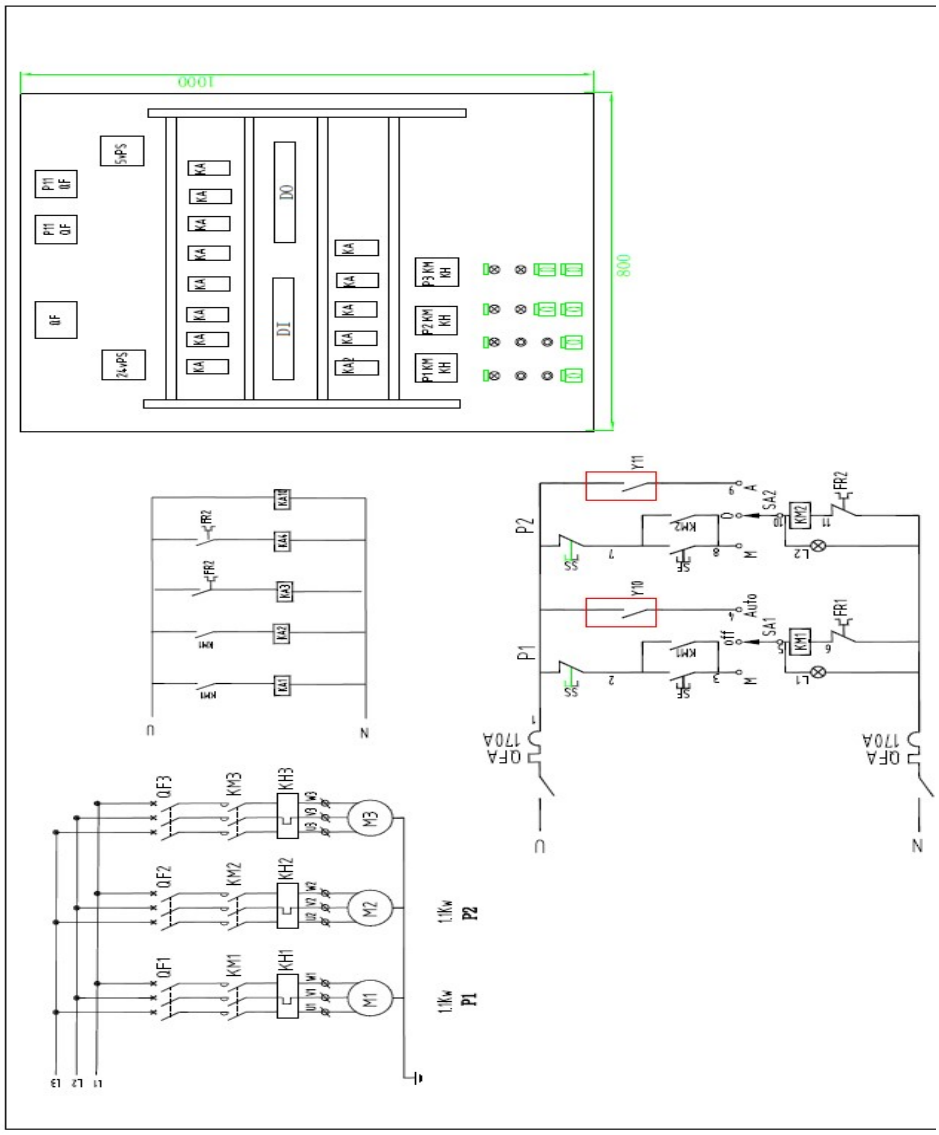
2- عند تصميم اللوحة الكهربائية ، نوصى بمراعاه أبعاد الادوات الكهرومغناطيسية مثل الكنتاكتر لتجنب الضجيج .

3- استخدام برتوكول profibus مباشرة والاستغناء عن برتوكول modbus ، نسبتا للمميزات العالية لبرتوكول profibus .

References المراجـــع

[1] siemens simutie configuration hard ware and communication setp7 manual,edition 03\2006.
[2] profibus technology and application,system description version october2002.
[3] spc3 seimens profibus controller,hardware description09/25/2.
[4] lingxia lui,liang song,xiang dong chen.the development Design of the intelelligent slav station based an the profibus- DP2008 international conference on computer science.
[5] modbus over serial line Specification & implementation guide v1.0.
[6] http://www.rodbus.org
[7] design of profibus intelligent slave bassed control system tiangin university of technology china2011- prof.song -mohaned kamal.

الملاحق



شکل(6-1)electric panel design designed by auto coa

DO firmware

1-Main file.c

```
//ICC-AVR application builder // Target : M128
//                Crystal:                 8.0000Mhz
                                                   //
#include                                  <iom128v.h>
#include                                   <macros.h>
#define          uchar          unsigned          char
#define        uint               unsigned         int
#define                  PLC_ADDR                    2
             ////////////////////////////////////////

#include                   "ATEMGA_128_SERVER_DEFINE.H"
#include                     "ATEMGA_128_SERVER_SEND.H"
void                                port_init_Y(void)
                                                    {
 PORTA                       =                    0xFF;
 DDRA                              =              0xFF;
 PORTB                       =                    0xFF;
 DDRB                              =              0xFF;
 PORTC              =              0xFF;
 DDRC                              =              0xFF;
 PORTD                       =                    0xFF;
 DDRD                              =              0xFB;
 PORTE                       =                    0xFF;
 DDRE                              =              0xFF;
 PORTF                       =                    0xFF;
 DDRF                              =              0xFF;
 PORTG                       =                    0xFF;
 DDRG                              =              0xFF;
                                                    }

//call this routine to initialize all peripherals
void                                    init_devices(void)
                                                    {
                 //stop errant interrupts until set up

 CLI();         //disable         all         interrupts
 XDIV             =      0x00;      //xtal      divider
 XMCRA       =       0x00;       //external       memory
                                          port_init_Y();
                                            uart_init();
 MCUCR                       =                    0x00;
 EICRA      =      0x00;      //extended     ext      ints
 EICRB      =      0x00;      //extended     ext      ints
 EIMSK                       =                    0x00;
 TIMSK     =     0x00;     //timer     interrupt     sources
 ETIMSK     =     0x00;     //extended     timer     interrupt
                                                  sources
 SEI();              //re-enable              interrupts
 //all     peripherals     are     now     initialized
```

```
        //Watchdog initialize

        void watchdog_init(void)
        {
         WDR (); //this prevents a timeout on enabling
         WDTCR |= (1<<WDCE) | (1<<WDE);/* 30-Oct-2006 Umesh*/
         WDTCR = 0x0F; //WATCHDOG ENABLED - dont forget to
        issue WDRs
        }

        void main(void)
        {
         uint f=0;
         init_devices();

         DelayMs(50);
         while(1)
         {
        RECEIVE_485_XY();
        watchdog_init();
         }
        }
        2- server_define.h
uchar                                                   i,
                                                      j_plc,
                                              PLC_Y_DATA[4],
                                            SEND_BUS_MOD_Y[20],
                                             MODBUS_DATA_R[20],
                                                    check_H,
                                                    check_L;

uint                                                    CRC;

//UART0                                              initialize
//        desired        baud        rate:        9600
//          char          size:          8          bit
//                  parity:                    Disabled
void                                        uart_init(void)
                                                        {
 UCSR1B  =  0x00;  //disable  while  setting  baud  rate
 UCSR1A    =    0x00;//not    double    the    baud.
 UCSR1C                     =                    0x06;
//              UBRR1L              =              0x10;
 UBRR1L   =   0x33;   //set   baud   rate   lo.baud=9600
 UBRR1H    =    0x00;    //set    baud    rate    hi
 UCSR1B                     =                    0x18;
                                                    }

void                DelayMs(uint                  i_del)
{char                                             j_del;
                                  for(;i_del!=0;i_del--)
```

```
{for(j_del=1000;j_del!=0;j_del--){;}
}
}

void send_RS485(uchar i_RS485)
{
DelayMs(4);

while(!(UCSR1A&(1<<UDRE1)));
UDR1=i_RS485;
DelayMs(4);
}


unsigned int CRC_check(unsigned char *fp_CRC,unsigned
char count)
  {
     unsigned int CRC=0XFFFF;
     unsigned char CRC_count;
     for(CRC_count=0;CRC_count<count;CRC_count++)
     {
        CRC=CRC^*(fp_CRC+CRC_count);//used the lower
8bit Xor the first number
        for(i=0;i<8;i++)//make the reg_CRC  right
shift 8 times
        {
            if(CRC&1)//if the bit you remove is 1,use
the reg_CRC Xor the 0XA001.
            {
                CRC>>=1;
                CRC^=0XA001;
             }
                else CRC>>=1;
          }
       }
        return CRC;
     }
```

3- server_send.h

```
void                                        INPUT_Y_ACTIVE()
                                                        {
                                             PORTA&=~(1<<2);
          PLC_Y_DATA[0]=~((PORTA&0X03)+(PORTE&0XFC));
                                    PLC_Y_DATA[1]=~PORTF;
    PLC_Y_DATA[2]=~((PORTB&0X01)+((PORTB&0XFC)>>1)+((
                                          PORTG&0X08)<<4));
                                PLC_Y_DATA[3]=~(PORTC);
                                                        }

void                                          SEND_RS485_Y()
                                                        {
                             SEND_BUS_MOD_Y[0]=PLC_ADDR;
                                 SEND_BUS_MOD_Y[1]=0X01;
 SEND_BUS_MOD_Y[2]=0X04;
                       SEND_BUS_MOD_Y[3]=PLC_Y_DATA[0];
```

```
                        SEND_BUS_MOD_Y[4]=PLC_Y_DATA[1];
                        SEND_BUS_MOD_Y[5]=PLC_Y_DATA[2];
                        SEND_BUS_MOD_Y[6]=PLC_Y_DATA[3];

                        CRC=CRC_check(SEND_BUS_MOD_Y,7);
                                    check_H=(CRC&0X00FF);
 check_L=(CRC>>8);//change   the   high   8bits   with
the              lower              8bits

                              SEND_BUS_MOD_Y[7]=check_H;
                              SEND_BUS_MOD_Y[8]=check_L;
                                        for(i=0;i<=8;i++)
                        {send_RS485(SEND_BUS_MOD_Y[i]);}
                                        for(i=0;i<=8;i++)
                          SEND_BUS_MOD_Y[i]=0;//±£ÁôÇ°Öµ

}

void                                      RECEIVE_485_XY()
                                                        {
                                  while(!(UCSR1A&(1<<RXC1)));
                                       MODBUS_DATA_R[0]=UDR1;
                                if(MODBUS_DATA_R[0]==PLC_ADDR)
                                                        {
                                        for(i=1;i<8;i++)
                                                        {
                                  while(!(UCSR1A&(1<<RXC1)));
                                       MODBUS_DATA_R[i]=UDR1;
                                                        }
     if(MODBUS_DATA_R[1]==0x01)//MODBUS   Y   REG   ¶ÁY
                                                        {
                        CRC=CRC_check(MODBUS_DATA_R,6);
                                    check_H=(CRC&0X00FF);
     check_L=(CRC>>8);//change    the    high    8bits
with          the          lower          8bits

      if(MODBUS_DATA_R[6]==check_H&&MODBUS_DATA_R[7]==c
                                             heck_L)//CRC
                                       {INPUT_Y_ACTIVE();
                                          DelayMs(35);
       SEND_RS485_Y();                                  }
                                                      }
   if(MODBUS_DATA_R[1]==0x05)//MODBUS    y7    REG
                                                        {
                        CRC=CRC_check(MODBUS_DATA_R,6);
                                    check_H=(CRC&0X00FF);
    check_L=(CRC>>8);//change   the   high   8bits   with
the              lower              8bits

      if(MODBUS_DATA_R[6]==check_H&&MODBUS_DATA_R[7]==c
                                             heck_L)//CRC
                                    {char h_lo,h_hi,check;
        h_hi=MODBUS_DATA_R[2];h_lo=MODBUS_DATA_R[3];
                                          check=MODBUS
```

```
                                                    _DATA_R[4];
                                                  if(h_hi==0x00)
                                                                {
          if(h_lo==0x00){if(check==0XFF){PORTA&=(0XFF-
                                                    0X01);}else
           if(check==0X00){PORTA=PORTA|0X01;}}//00--PINA0
          if(h_lo==0x01){if(check==0XFF){PORTA&=(0XFF-
                                                    0X02);}else
           if(check==0X00){PORTA=PORTA|0X02;}}//01--PINA1
          if(h_lo==0x02){if(check==0XFF){PORTE&=(0XFF-
                                                    0X04);}else
           if(check==0X00){PORTE=PORTE|0X04;}}//02--PINE2
          if(h_lo==0x03){if(check==0XFF){PORTE&=(0XFF-
                                                    0X08);}else
           if(check==0X00){PORTE=PORTE|0X08;}}//03--PINE3
          if(h_lo==0x04){if(check==0XFF){PORTE&=(0XFF-
                                                    0X10);}else
           if(check==0X00){PORTE=PORTE|0X10;}}//04--PINE4
          if(h_lo==0x05){if(check==0XFF){PORTE&=(0XFF-
                                                    0X20);}else
           if(check==0X00){PORTE=PORTE|0X20;}}//05--PINE5
          if(h_lo==0x06){if(check==0XFF){PORTE&=(0XFF-
                                                    0X40);}else
           if(check==0X00){PORTE=PORTE|0X40;}}//06--PINE6
          if(h_lo==0x07){if(check==0XFF){PORTE&=(0XFF-
                                                    0X80);}else
           if(check==0X00){PORTE=PORTE|0X80;}}//07--PINE7

            if(h_lo==0x08){if(check==0XFF){PORTF&=(0XFF-
                                                    0X01);}else
           if(check==0X00){PORTF=PORTF|0X01;}}//10--PINF0
          if(h_lo==0x09){if(check==0XFF){PORTF&=(0XFF-
                                                    0X02);}else
           if(check==0X00){PORTF=PORTF|0X02;}}//11--PINF1
          if(h_lo==0x0A){if(check==0XFF){PORTF&=(0XFF-
                                                    0X04);}else
           if(check==0X00){PORTF=PORTF|0X04;}}//12--PINF2
          if(h_lo==0x0B){if(check==0XFF){PORTF&=(0XFF-
                                                    0X08);}else
           if(check==0X00){PORTF=PORTF|0X08;}}//13--PINF3
          if(h_lo==0x0C){if(check==0XFF){PORTF&=(0XFF-
                                                    0X10);}else
           if(check==0X00){PORTF=PORTF|0X10;}}//14--PINF4
          if(h_lo==0x0D){if(check==0XFF){PORTF&=(0XFF-
                                                    0X20);}else
           if(check==0X00){PORTF=PORTF|0X20;}}//15--PINF5
          if(h_lo==0x0E){if(check==0XFF){PORTF&=(0XFF-
                                                    0X40);}else
           if(check==0X00){PORTF=PORTF|0X40;}}//16--PINF6
          if(h_lo==0x0F){if(check==0XFF){PORTF&=(0XFF-
                                                    0X80);}else
if(check==0X00){PORTF=PORTF|0X80;}}//17--PINF7
          if(h_lo==0x10){if(check==0XFF){PORTB&=(0XFF-
                                                    0X01);}else
           if(check==0X00){PORTB=PORTB|0X01;}}//20--PINB0
```

```
        if(h_lo==0x11){if(check==0XFF){PORTB&=(0XFF-
                                                0X04);}else
         if(check==0X00){PORTB=PORTB|0X04;}}//21--PINB2
        if(h_lo==0x12){if(check==0XFF){PORTB&=(0XFF-
                                                0X08);}else
         if(check==0X00){PORTB=PORTB|0X08;}}//22--PINB3
        if(h_lo==0x13){if(check==0XFF){PORTB&=(0XFF-
                                                0X10);}else
         if(check==0X00){PORTB=PORTB|0X10;}}//23--PINB4
        if(h_lo==0x14){if(check==0XFF){PORTB&=(0XFF-
                                                0X20);}else
         if(check==0X00){PORTB=PORTB|0X20;}}//24--PINB5
        if(h_lo==0x15){if(check==0XFF){PORTB&=(0XFF-
                                                0X40);}else
         if(check==0X00){PORTB=PORTB|0X40;}}//25--PINB6
        if(h_lo==0x16){if(check==0XFF){PORTB&=(0XFF-
                                                0X80);}else
         if(check==0X00){PORTB=PORTB|0X80;}}//26--PINB7
        if(h_lo==0x17){if(check==0XFF){PORTG&=(0XFF-
                                                0X08);}else
         if(check==0X00){PORTG=PORTG|0X08;}}//27--PING3
        if(h_lo==0x18){if(check==0XFF){PORTC&=(0XFF-
                                                0X01);}else
         if(check==0X00){PORTC=PORTC|0X01;}}//30--PINC0
        if(h_lo==0x19){if(check==0XFF){PORTC&=(0XFF-
                                                0X02);}else
         if(check==0X00){PORTC=PORTC|0X02;}}//31--PINC1
        if(h_lo==0x1A){if(check==0XFF){PORTC&=(0XFF-
                                                0X04);}else
         if(check==0X00){PORTC=PORTC|0X04;}}//32--PINC2
        if(h_lo==0x1B){if(check==0XFF){PORTC&=(0XFF-
                                                0X08);}else
         if(check==0X00){PORTC=PORTC|0X08;}}//33--PINC3
        if(h_lo==0x1C){if(check==0XFF){PORTC&=(0XFF-
                                                0X10);}else
         if(check==0X00){PORTC=PORTC|0X10;}}//34--PINC4
        if(h_lo==0x1D){if(check==0XFF){PORTC&=(0XFF-
                                                0X20);}else
         if(check==0X00){PORTC=PORTC|0X20;}}//35--PINC5
        if(h_lo==0x1E){if(check==0XFF){PORTC&=(0XFF-
                                                0X40);}else
         if(check==0X00){PORTC=PORTC|0X40;}}//36--PINC6
        if(h_lo==0x1F){if(check==0XFF){PORTC&=(0XFF-
                                                0X80);}else
         if(check==0X00){PORTC=PORTC|0X80;}}//37--PINC7
                                              DelayMs(40);
                                        for(i=0;i<8;i++)
{send_RS485(MODBUS_DATA_R[i]);}//Send     to     PC
                                        for(i=0;i<8;i++)
                                    {MODBUS_DATA_R[i]=0;}
                                                    }
                                    for(i=0;i<=7;i++)       }
                                        {MODBUS_DATA_R[i]=0;}
```