

تصميم جهاز
ملاحقة الاشعاع
الشمسي

January 5

2011

إعداد: م.أسامة الفاضل

www.kawngroup.com

ملخص المشروع :

ما تزال الطاقة الشمسية تحظى بسمعة ضعيفة كوسيلة من وسائل التوسع في استخدام الطاقات المتجددة كطاقة بديلة. عند استخدام نظام ملاحقة شمسية سنضمن انتاج كمية أكبر من الطاقة الكهربائية وذلك بسبب بقاء مصفوفة الخلايا الشمسية على محاذات من أشعة الشمس طوال فترة سطوعها. إن هذا المشروع الذي تم تصميمه في جامعة ولاية كليفلاند الأمريكية تضمن تصميم وإنشاء نظام ملاحقة شمسية يعتمد على المتحكمات الميكروية. تم معالجة النظام ومكوناته العملية بشكل كامل وايضا تم تقييم التصميم نظرياً وشرح مبدأ عمله. وأخيراً ناقش المشروع المشاكل التي يمكن أن نواجهها وبعض الحلول البديلة والتحسينات التي يمكن القيام بها.

مقدمة :

إن حلول الطاقات المتجددة تصبح ذات شهرة أكبر وكفاءة أعلى بشكل متزايد مع الزمن. والخلايا الكهروضوئية ليست إلا مثال على ذلك. إن رفع قيمة خرج الخلايا الشمسية إلى قيمة عظمى هو أمر مرغوب به وذلك لرفع المردود للخلايا الشمسية إلى قيمة أعلى. أحد الطرق للحصول على كمية اشعاع أكبر هي ان تكون الخلايا الشمسية بمواجهة الشمس دوماً باستخدام ما يسمى نظام ملاحقة شمسي (نظام التتبع الشمسي). هذا الحل يعتبر غالباً اقتصادياً أكثر بكثير من شراء عدد أكبر من الخلايا من أجل الحصول على خرج أعلى. ولقد تم التقدير بأنه حصيلته ما تولده الخلايا الشمسية يمكن أن يزداد بين قيم تتراوح بين الـ ٣٠ و الـ ٦٠ % عندما نستخدم أجهزة ملاحقة شمسية عن القيمة التي نحصل عليها عند استخدام خلايا ثابتة. ويتناول هذه المشروع تصميم أحد هذه الأنظمة للحصول على أعظم مردود.

ويبدأ هذا المشروع بعرض خلفية عن الحساسات الضوئية والمحركات الخطوية من ناحية تتعلق بالمشروع. ويتابع المشروع بشرح منهجية تصميم معينة تتعلق بـ الخلايا الضوئية، المحركات الخطوية ودارات القيادة لها، اختيار المتحكمات المايكروية لتنظيم التوتر، البنية الفيزيائية، وشرح عن نظام العمل البرمجي المستخدم في النظام. ويخرج المشروع بتوصيات من أجل تطوير هذا النظام إلى نظام مستقبلي أفضل.

معلومات أساسية :

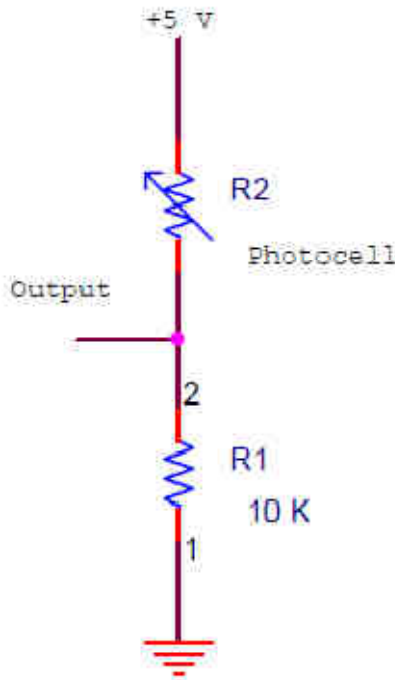
قبل الحديث عن تصميم جهاز الملاحقة الشمسية لابد من التعرف على بعض المعلومات الأساسية عن المكونات والأجزاء الألكترونية التي سيتم استخدامها. مثل المحركات الخطوية والحساسات الضوئية وذلك لتوفير معلومات أكبر عن كيفية تأثير هذه الأنظمة على نظام الملاحقة الشمسي الكلي.

١ - الحساسات الضوئية:

الحساسات الضوئية هي من أكثر أنواع الحساسات انتشاراً. و أبسط أنواع الحساسات الضوئية هو عبارة عن مقاومة ضوئية، التي يمكن أن تكون من نوع كبريتيد الكادميوم (CdS) أو زرنيخ الغاليوم (Ga As). أما الأنواع الأكثر تعقيداً فهي عبارة عن ديود ضوئي متبع بترانزستور ضوئي.

جهاز التتبع الشمسي يستخدم خلية ضوئية من نوع كبريتيد الكادميوم من أجل التحسس للضوء وهذا النوع من الحساسات هو الأرخص ثمناً والأقل تعقيداً، هذا النوع من الخلايا هي من العناصر السالبة التي مقاومتها تتناسب عكساً مع كمية الضوء الساقط عليه. ومن أجل اعداد هذه الخلية توصل على التسلسل مع مقاومة. وبالتالي ينشأ لدينا مجزء كمون والخرج عند الوصلة يحدد عندها من مجموع المقاومتين. الشكل التالي يوضح الدارة المكافئة.

في هذا المشروع كان الرغبة هي رفع توتر الخرج عند زيادة شدة الضوء لذا تم وضع الخلية الضوئية في الأعلى.



٢ - المحرك الخطوي وطريقة التحكم به:

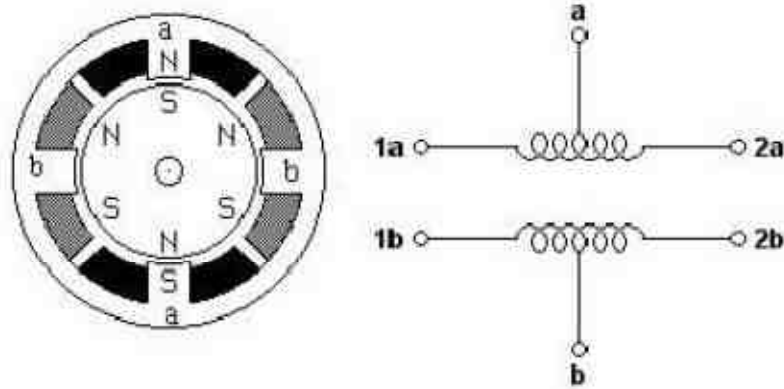
يستخدم عادة المحرك الخطوي من أجل تطبيقات التحريك الدقيقة. وجميع هذه المحركات تمتاز بخمس صفات أساسية مشتركة والتي تجعلها المثالية من أجل مثل هذه العملية.

- ليس لها مسفرات.
- لا تتعلق بالحمل.
- لها امكانية التموضع والتوقف عند اي زاوية.
- عزم كبج قوي .
- خصائص استجابة ممتازة.

ان هذه المحركات لها ثلاثة أنواع وهي: ذات المغناط الدائمة، ذات الممانعة المتغيرة والنوع الهجين.

إن طريقة ترتيب الملفات على الجزء الثابت هي الشيء الذي يميز كل نوع من الأنواع الثلاثة. المحركات ذات الأقطاب الدائمة ممكن أن تملك ملفات أحادية القطب أو ملفات ثنائية القطبية كمغناط.

عادة تستخدم أجهزة التتبع الشمسي محركات خطوية ذات قطب واحد " أحادية القطبية " وسيقتصر الشرح على هذا النوع من المحركات. المحركات الأحادية القطبية تملك وشيعتان ولك منهما نقطة وسط كما هو مبين بالشكل.



يتم ربط نقطة الوسط إلى طرف تغذية موجب بينما يتم ربط نهايتي الوشيعة إلى الطرف الأرضي من أجل تشكيل حقلين كهربائيين متعاكسين في كل منتصف من الوشيعة.

الشكل التالي يوضح طريقة إعطاء التغذية لمحرك خطوي ذو أربع خطوات أن عدد الخطوات يساوي ضعف عدد الوشائع ويتم تدوير المحرك عن طريق تطبيق توتر على الوشائع بتتابع كالموضح بالجدول التالي:

Index	1a	1b	2a	2b
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0
7	0	0	1	0
8	0	0	0	1

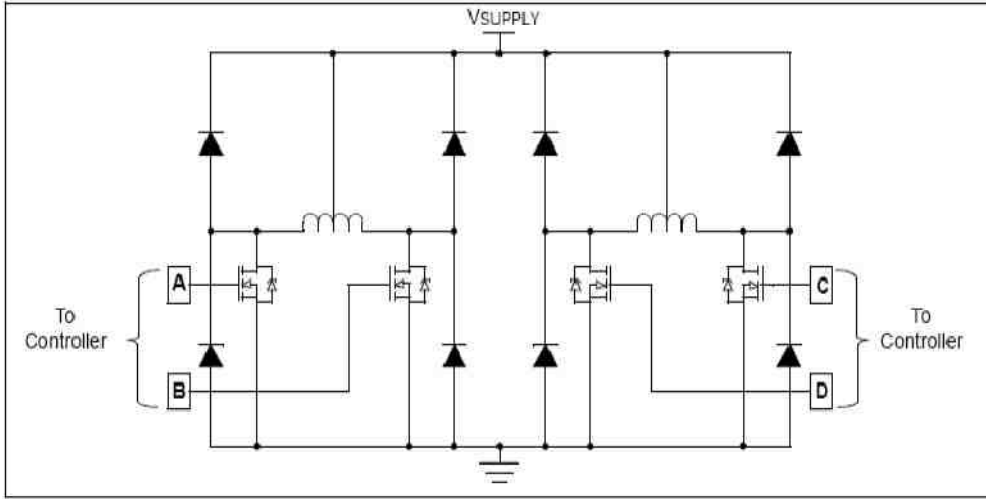
Clockwise Rotation ↓

المحرك يمكن أن يكون نصف خطوي. والتحرك النصف خطوي ممكن أن يتم بتغذية وشيعة واحدة وبعدها وشيعتين ومن ثم وشيعة واحدة أيضاً. بتتابع كالمعروض في الأسفل هذا المشروع يستخدم محرك نصف خطوي ويتم مناقشته بشكل تفصيلي لاحقاً في هذا البحث. ويبين الجدول التالي طريقة القيادة كنصف خطوة.

Index	1a	1b	2a	2b
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1
9	1	0	0	0
10	1	1	0	0
11	0	1	0	0
12	0	1	1	0
13	0	0	1	0
14	0	0	1	1
15	0	0	0	1
16	1	0	0	1

Half Step Sequence

وأخيراً نحتاج إلى دارة من أجل قيادة المحرك الخطوي. والدارة الأساسية من أجل مثل هذه العملية من التحكم مبين بالشكل التالي أما الدارة الخاصة من أجل هذا التطبيق فهي موضحة ومشروحة لاحقاً.



٣- الية بناء المشروع :

سنناقش في هذا القسم الطريقة التي سيتم وفقها بناء جهاز التعقب الشمسي وقد تم تجزأة هذا المشروع لقسمين وذلك لكي تكون عملية التصميم أكثر سهولة. الخطوة العملية الأولى في المشروع هي قراءة عدة قيم عن طريق الحساسات الضوئية ومقارنة هذه القيم ثم يتم اعطاء أوامر للمحرك ليقيم بتحريك النظام حسب القيم العظمى المقروءة من قبل الحساسات والتي تقابل اتجاه الاشعاع الشمسي الأعظمي وبالتالي جهة الشمس. وباقي الأقسام من المشروع معنية بدراسة الاعتبارات لاختيار البرامج والقطع المستعملة الأنسب لتنفيذ المشروع.

١-٣ تصميم الحساسات الضوئية :

كما تم الشرح سابقاً فإن جهاز التعقب الشمسي يستخدم خلايا ضوئية من نوع كبريتيد الكاديوم من أجل تحديد الشدة الضوئية. ومقاومة متممة قيمتها تساوي الـ ١٠ كيلو أوم وتستعمل من أجل اكمال الدارة الموضحة بالشكل السابق. في هذه الطريقة من التركيب توتر الخرج سوف يتزايد مع ازدياد الاشعاع الضوئي .

أن قيمة المقاومة المتممة يجب ان تختار على أساس تحقيق أوسع مقدار من قيم توترات الخرج. إن قيمة مقاومة للخلية الضوئية التي تم قياسها تبعاً لنوع الاضاءة والتي هي : الظروف المعتمدة ، الاضاءة المتوسطة ، الاضاءة الساطعة معطاة بالجدول التالي:

نوع الاضاءة	المقاومة المقاسة
الظلام " وضع قماش أسود على الخلية الضوئية "	٥٠ كيلو أوم
متوسطة " الاضاءة العادية في الغرفة "	٤.٣٥ كيلو أوم

الساطعة " توجيه مصباح ضوئي على الخلية "	٢٠٠ أوم
---	---------

وفي حال اختيار مقاومة متممة ١٠ كيلو اوم سوف ينتج لدينا قيم توترات كالتالي :

$$\text{Minimum} = 5 \text{ V} \times (10 \text{ K}\Omega / (10 \text{ K}\Omega + 50 \text{ K}\Omega)) = 0.83 \text{ V}$$

$$\text{Maximum} = 5 \text{ V} \times (10 \text{ K}\Omega / (10 \text{ K}\Omega + 4.35 \text{ K}\Omega)) = 3.48 \text{ V}$$

لذا فإن تأرجح قيمة الخرج هو ٢.٦٥ فولت. وهذه القيمة ليست مثالية ولكنها تعتبر كافية ولا نحتاج إلى قيم أكبر في هذا المشروع .

٢-٣ المتحكم المايكروبي :

بما أن المشروع يركز على نظام التحكم المتكامل فإن المتحكم المايكروبي هو من أساس النظام. أن المتحكم المستخدم في هذا المشروع يجب أن يحول التوتر التماثلي من خرج الخلية الضوئية إلى قيم رقمية وأن يوفر أربع مخارج من أجل التحكم بدوران المحرك. المتحكم من النوع PIC16F877 قد تم اختياره لأنه يلبي هذه الحاجات وبالإضافة إلى انه كثير التواجد وسهل الاستخدام، وأيضاً يمتلك هذا المتحكم الميزات الثلاثة التالية والتي تساعد على تحقيق الهدف من المشروع :

١- محول متعدد القنوات من تماثلي إلى رقمي ذو ١٠ بت.

٢- ٥ مداخل ومخارج.

٣- ذاكرة من نوع EEPROM ذات ٢٥٦ x ٨ بايت.

يتم وصل مهتز كرسنالي مع المتحكم المايكروبي من أجل توفير نبضات التوقيت المطلوبة. هذه السرعة هي كافية من أجل هذا التطبيق وبين الشكل التالي مخطط تفصيلي للمتحكم PIC16F877 .

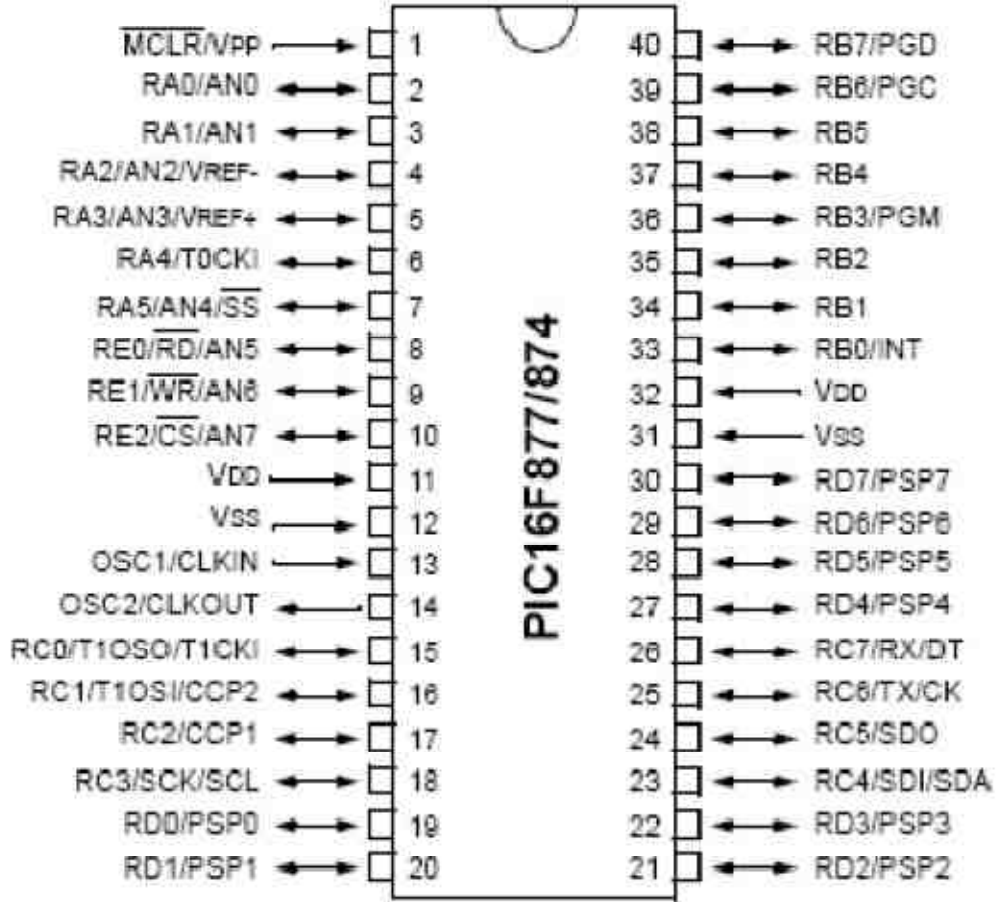


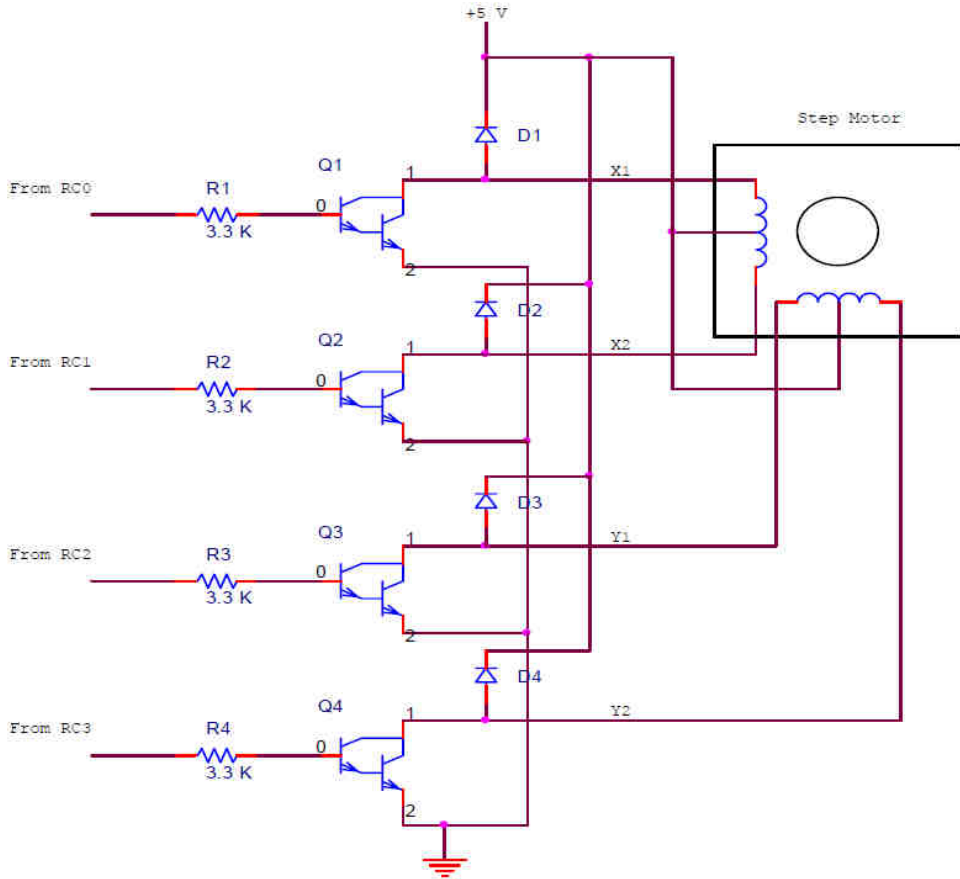
Figure 6 – PIC16F877 Pinout

٣-٣ دائرة القيادة والمحرك الخطوي:

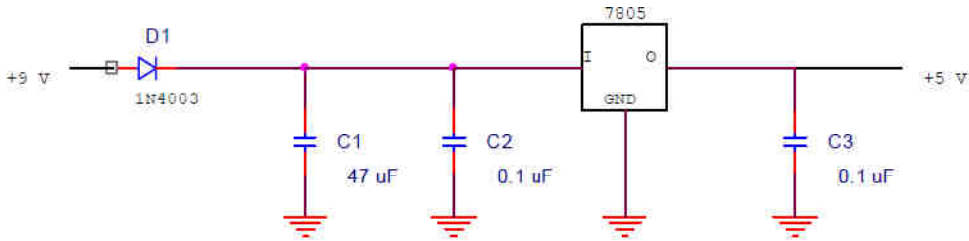
لقد تم اختيار محرك خطوي وحيد القطبية من أجل تحريك حساسات نظام الملاحقة. وقد ذكرنا سابقاً فان اختيار المحرك الخطوي كان بسبب دقة في العمليات التي تحتاج إلى التحريك الدقيق، بالإضافة إلى ذلك فإن دارت التحكم ليست معقدة في حال كون المحرك وحيد القطبية. المحرك الذي استخدم في هذا المشروع هو عبارة عن محرك ذو دخل ٥ فولت وخطوة بمقدار ٧.٥ درجة وهو رباعي الأشواط وأحادي القطبية. كما أن المحركات النصف الخطوية تحقق دقة أكثر في التحريك. وهذا يؤدي إلى ٣.٧٥ درجة بالخطوة. وتتابع العمل المستخدم موضح بالشكل التالي:

Index Position	Y2	Y1	X2	X1
1	0	1	0	1
2	0	0	0	1
3	1	0	0	1
4	1	0	0	0
5	1	0	1	0
6	0	0	1	0
7	0	1	1	0
8	0	1	0	0

والمخطط التالي يوضح دارة قيادة المحرك الخطوي، إن هذا التصميم وبرنامج قيادة المحرك المرافق لهذا التصميم معتمد على برنامج لاب ٨. ترانزستورات دارلينغتون تم اختيارها من أجل أن تستخدم مع دارة قيادة المحرك. كل ترانزستور موصول مع مقاومة ٣.٣ كيلو أوم إلى خرج يستخدم من أجل وصل أو فصل التيار إلى وشائع المحرك. هذه يعطي التتابع المناسب من أجل تدوير المحرك مع أو عكس عقارب الساعة. الديودات التي تصل المجمعات إلى التغذية الموجبة تحمي الترانزستورات من التيارات العكسية.



إن هذا المتحكم يحتاج إلى منظم توتر ذو النوع 7805 والذي يثبت قيمة التوتر على الـ ٥ فولت ويوضح الكشل التالي دراة مستخدمة من أجل تنظيم التوتر والتي تقوم بتحويل توتر منبع ذو قيمة ٩ فولت إلى قيمة ٥ فولت لتغذية المتحكم.



٤- الإنشاء:

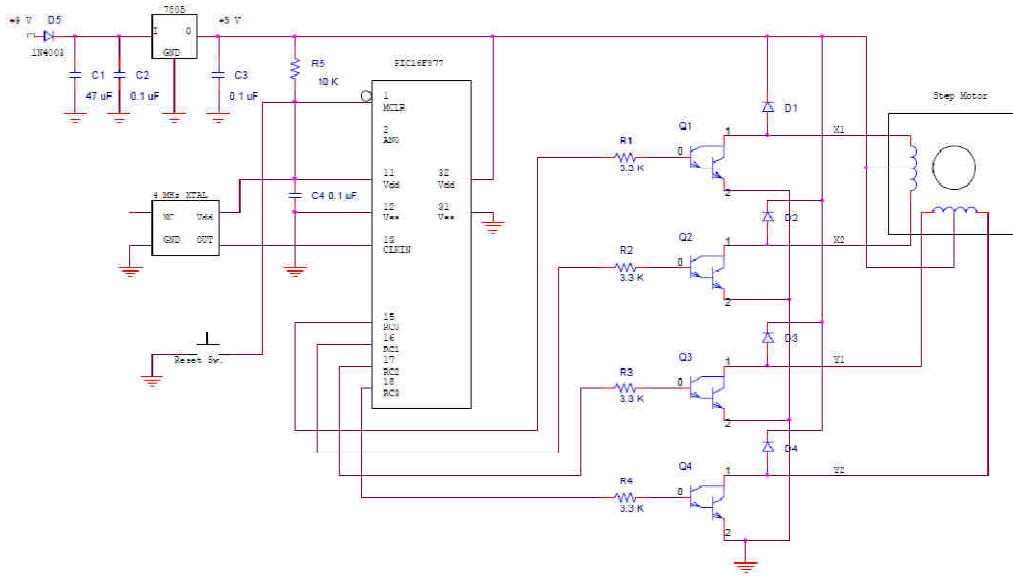
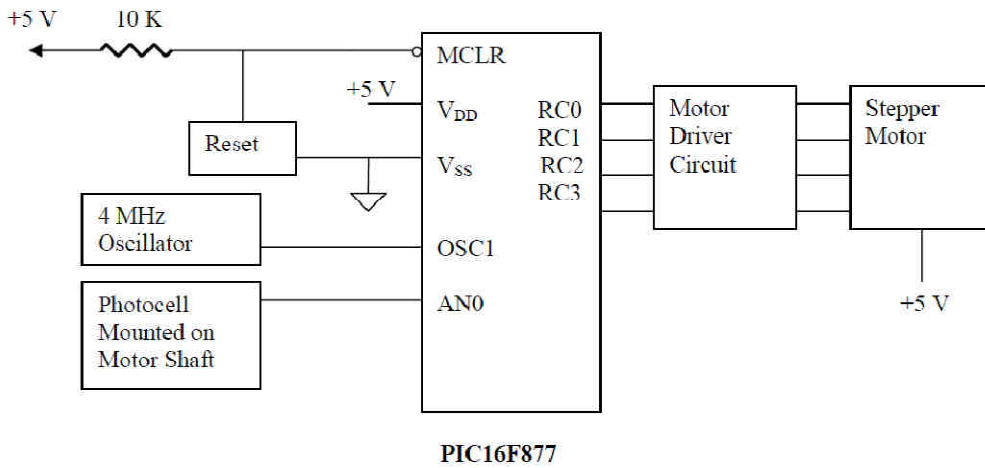
أخيراً فإن جميع الأجزاء الثانوية المذكورة في الفقرة ٣-١ إلى الفقرة ٣-٣ هي عبارة عن أجزاء مكملة لعمل النظام الشكل التالي يبين الشكل النهائي للمشروع بينما يبين الشكل الذي يليه الشكل الكامل لجميع التجهيزات المستخدمة في المشروع.

بعض تفاصيل الإنشاء التي تستحق الذكر والتي تتعامل مع المحرك والخلية الضوئية:

المحرك ركب على لوح بلاستيكي مثقب باستخدام براغي معدنية من أجل الحصول على تثبيت متين.

الخلايا الضوئية تم تثبيتها على لوح خشبي والذي وصل بدوره إلى عمود المحرك.

وفي النهاية تم اضافة كبسة اعادة الوضع من أجل السماح للمتحكم المايكروبي للعودة إلى وضعه البدائي بعد أن يدخل في وضعية القيلولة.



ويوضح الجدول التالي جميع الأجزاء المستخدمة في المشروع :

Item	Size or Part No.	Qty.
Microchip microcontroller	PIC16F877	1 ea.
Oscillator, crystal	4 MHz	1 ea.
Voltage regulator	7805	1 ea.
Photocell	Cadmium sulfide	1 ea.
Step motor, unipolar	5 V	1 ea.
Capacitor	0.1 μ F	3 ea.
Capacitor	47 μ F	1 ea.
Resistor	10 K Ω	1 ea.
Resistor	3.3 K Ω	4 ea.
Diode	1N4003	4 ea.
Transistor, Darlington	2SD1276A	4 ea.
Switch, momentary, pushbutton	Normally open	1 ea.
Motor/sensor mounting accessories	Various	Various

٥- البرنامج وطريقة عمل النظام:

أن طريقة عمل المشروع السابق تعتمد بشكل أساسي على لغة الألة وهي كان أكثر من مناسبة من أجل عمل المشروع وزيادة المعرفة والتعمق بلغات البرمجة. ويمكن تقسيم عمل البرنامج إلى أربعة أجزاء رئيسية والتي هي :

الجزء الأول والذي يتعامل مع التوضع:

قبل تغذية النظام فإن الخلايا الضوئية يجب أن تكون موجهة يدوياً إلى نقطة البداية والتي هي نقطة سطوع الشمس. وبعد أن يوجه النظام إلى هذه النقطة فإن حساسات نظام الملاحقة سوف تتحرك ٣.٧٥ درجة كخطوة بالثانية باتجاه عقارب الساعة حتى تصبح قيمة الاشعاع الضوئي أكبر من القيمة المعير عليها النظام مسبقاً. القيمة المعير عليها النظام تقابل توتر مرجعي ٤.٦٠ فولت. حيث اختيرت هذه القيمة لكي تتوافق مع توتر الخلية الكهروضوئية في حال كانت مغطاة وموجهة باتجاه أشعة الشمس. هذه القيمة تضمن أن نظام التتبع الشمسي سوف يلاحق قيمة عظمى للاشعاع الشمسي.

الجزء الثاني من البرنامج يتعامل مع الملاحقة الضوئية:

وهذا الجزء هو لب البرنامج، بعد أن يتم تعير الموضع البدائي لنظام الملاحقة بواسطة مصدر ذو شدة اضاءة عالية " الشمس " يصبح جاهزاً عندها لتعديل نفسه بدقة أكثر ومتابعة الضوء. المتابع يقيس في البداية شدة الضوء في المكان الموجود فيه وبعدها يتحرك بعكس عقارب الساعة بمقدار ٣.٧٥ درجة ثم يقوم بقياس قيمة الاشعاع عندها، بعدها يتحرك مع عقارب الساعة بمقدار ٣.٧٥

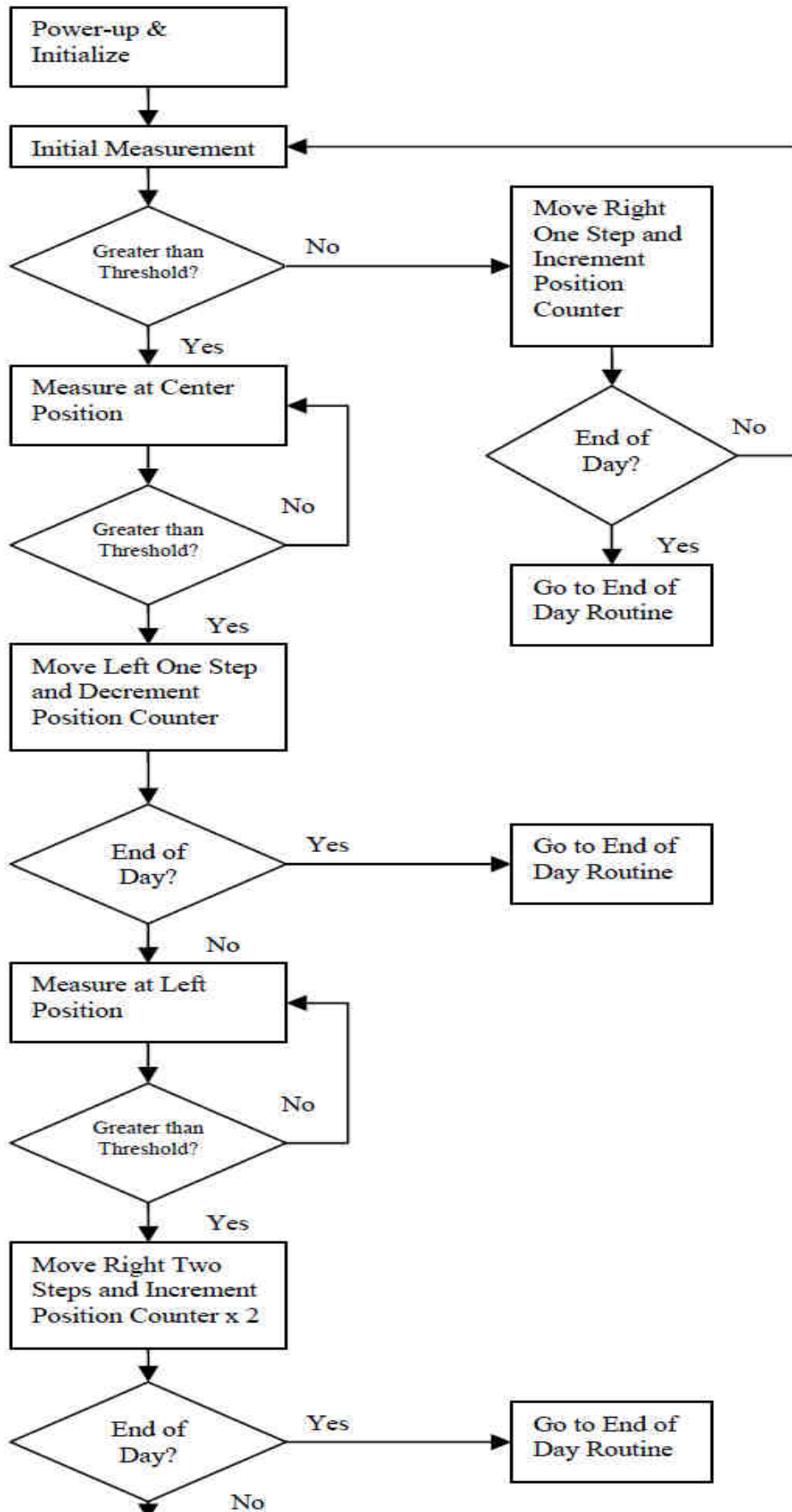
درجة ويأخذ قراءة أخيرة. ويقوم البرنامج بمقارنة القيمتين ومن ثم يتحرك باتجاه النقطة ذات قيمة الاشعاع الشمس الأكبر ويقف عندها. في حال تساوي القيمتين سوف يعود النظام إلى النقطة السابقة التي بدء عندها ويعيد عملية القياس من جديد وذلك بعد مضي أربعة دقائق قبل أن يقيس من جديد وهذه الزمن مأخوذ من حقيقة أن الشمس تتحرك بمقدار درجة واحدة كل أربعة دقائق .

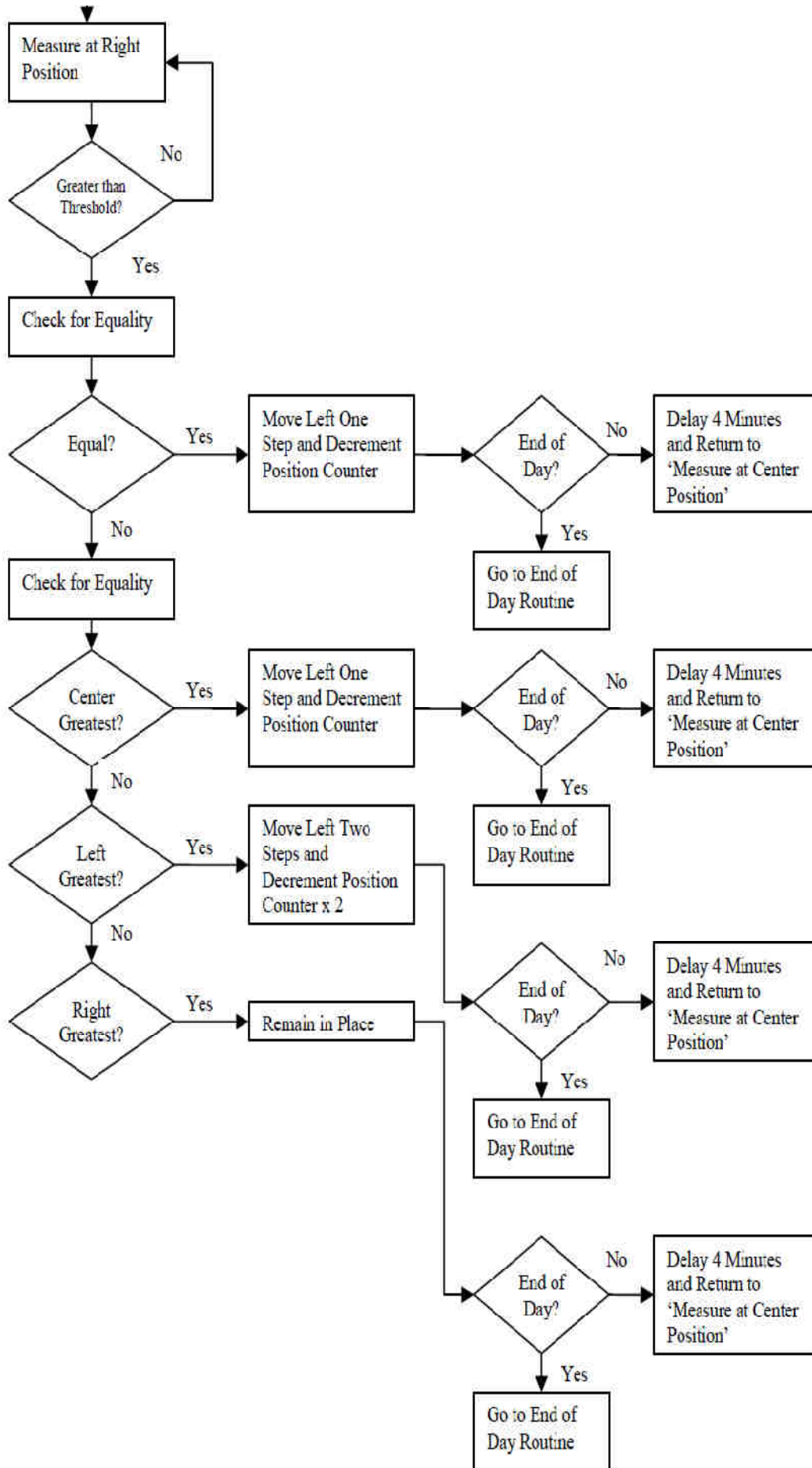
الجزء الثالث من البرنامج يتعامل مع التحسس للضوء الضعيف:

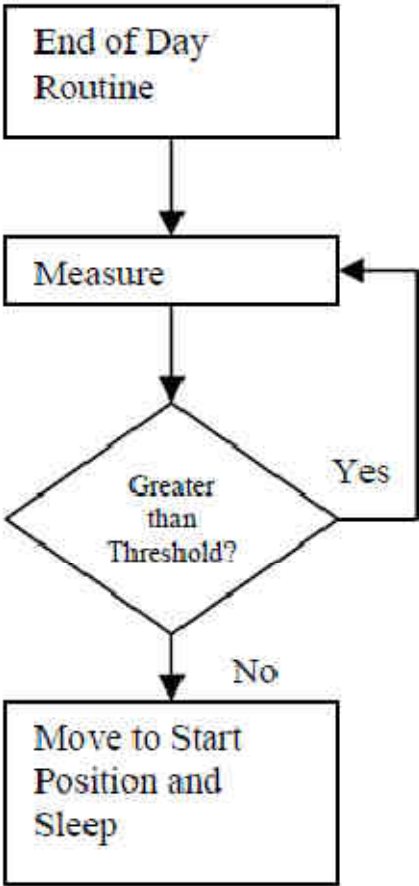
وهو يشابه العملية المذكورة بالجزء السابق، حيث اذا كانت كثافة الضوء أقل من عتبة الضوء الدنيا سوف يستمر الملاحق بالقياس في الموضع الذي هو فيه والتحرك حتى يصل إلى قياس يساوي عتبة الضوء الدنيا. وهذا القيمة في البرنامج اسندت لتقابل توتر مقداره ٣.٧٥ فولت. وهذه القيمة تتناسب مع القيمة التي تم قياسها من قبل خلية ضوئية مظلمة في يوم غائم وهي ثلاثم أوضاع الاشعاع في الايام الغائمة .

القسم الأخير من البرنامج يتيح لنظام المتابعة من أن يعيد نفسه في نهاية اليوم:

بعد كل حركة يقوم بها المحرك يتزايد أحد المتحولات أو يتناقص تبعاً من أجل معرفة الموقع الدقيق الذي وصل إليه النظام في أي وقت. بعد أن يقوم المتعقب بالدوران ١٨٠ درجة يتم التحقق من كثافة الضوء. اذا كانت قيمة كثافة الضوء تقابل قيمة توتر أقل من توتر العتبة الدنيا والتي هي ٣.٧٥ فولت كما ذكرنا سابقاً سيعود المتعقب إلى مكانه البدائي ويدخل في وضع الغفوة. واذا كانت القيمة أعلى من العتبة الدنيا سيستمر المتعقب بالقياس حتى تصل القيمة المقاسة أقل من العتبة الدنيا. وفي الوقت الذي يدخل فيه النظام حالة الغفوة يمكن اعادته إلى وضعه البدائي عن طريق مفتاح يدوي. والشكل التالي يوضح الخوارزمية التي تمثل عمل البرنامج.







٦- تحليل التصميم والنتائج :

ان البرنامج والتركيب للمشروع تم تقسيمها إلى مراحل أثناء تطوير النظام والية عمله ككل. وكانت هذه الأقسام هي عبارة عن التحسس للضوء ، قيادة المحرك ، برنامج الملاحقة ، والتحسينات البرمجية. إن تقسيم طريقة عمل البرنامج وتبسيطه جعلت البرنامج أقصر ويحتاج زمن أقل عند التنفيذ. وهذا البرنامج حقق الهدف المطلوب منه عند البدء بالمشروع. ولكن كانت المشكلة في العتاد الصلب " التجهيزات " . المشكلة الأولى كانت عند تحرك النظام وتأثيره على الأسلاك الوصلة بين الخلية الضوئية والمحرك والإلتواءات الناتجة على هذه الأسلاك أثناء الحركة. هناك سلكان موصلان إلى الخلية الضوئية ومثبتان على محور المحرك. حالما يتحرك المتعقب من ٣٠ إلى ٤٥ درجة تسبب هذه الأسلاك عزمًا معاكسًا يسبب انزلاقًا للمحرك. وهذا يسبب بدوره تموضع خاطئ للنظام. والحل لهذه المشكلة كان عن طريق تثبيت أسلاك الحساس الضوئي إلى الجزء الخلفي من الحساس أثناء حركة النظام.

المشكلة الثانية تتعامل مع الحساس الضوئي. حيث قد تم اكتشاف أن الخلية تحتاج إلى حاجب ضوئي من أجل تركيز الأشعة الساقطة عليها بشكل حزمة. وقد تم حل هذه المشكلة عن طريق احاطة الخلية بماسورة سوداء من أجل الحصول على قناة توصل الضوء إلى مركزها وتمنع الضوء من السقوط عليها من جميع الاتجاهات.

٧- تطوير الجهاز في المستقبل :

الهدف من المشروع كان الحصول على النتائج المطلوبة في الزمن المخصص له. ولهذا فإن العديد من التعديلات قابل للتطبيق على التصميم الأولي. وهذا ما يوضح أن هذا النظام هو نظام تجريبي ويمكن إجراء العديد من التعديلات عليه. والأفكار التالية هي بعض الأفكار القابلة للتطبيق في هذا المجال:

- ١- علاج لمشكلة التواء الأسلاك والمرتبطة بالخلايا الضوئية يمكن وضع زيت للتشحيم على المحور أو استخدام محور بعزم أكبر أو تقصير طول الأسلاك.
- ٢- زيادة مجال التحسس والدقة للنظام عن طريق استخدام خلايا ضوئية ذات حساسية أعلى كاستخدام حساسات ضوئية ذات دارات مضخمة للإشارة مما يتيح دقة أكثر في المتابعة.
- ٣- يمكن استخدام مصفوفة ترانزستورات من نوع دارلينغتون ذو رقم UCN5804 من أجل اقلال عدد العناصر المستخدمة في النظام.
- ٤- يمكن تطبيق هذا النظام على محورين من أجل زيادة الدقة في الملاحقة وزيادة المردود.

٨- الاستنتاجات :

هذا البحث يوضح طريقة بناء نظام ملاحقة شمسي من عن طريق استخدام نظام بمعالج مايكروبي. وهو أيضاً يوضح حل برمجي مناسب من أجل زيادة مردود الأنظمة الشمسية إلى قيمة عظمى عن طريق توجيه النظام إلى نقطة الاشعاع الشمسي الأعظمي ومن ثم العودة إلى الوضع البدائي بعد غياب الشمس من أجل يوم جديد.

References:

Solar Tracker
Bill Lane*
Department of Electrical and Computer Engineering
Cleveland State University
Cleveland, Ohio 44115
EEC 517
April 30, 2008

الملحق: برمجة المعالج المايكروبي:

```
list p=16f877
include "p16f877.inc"
__CONFIG_CP_OFF & _CPD_OFF & _LVP_OFF & _WDT_OFF &
_BODEN_OFF & _PWRTE_OFF & _XT_OSC
;***Initial assignments**
cblock 0x20
Direction ;Motor direction register (left or right)
Position0 ;Center reference position register
Position1 ;Left reference position register
Position2 ;Right reference position register
Time1 ;Timer register for 1 msec delay function
Time2 ;Timer register for 250 msec delay function
Time3 ;Timer register for 1 sec delay function
Time4 ;Timer register for 1 min delay function
PositionCount ;Register to store net value of steps taken
Temp ;Register for return to east decrement position loop
endc
Left equ d'2' ;Left direction = 2
Right equ d'1' ;Right direction = 1
Index1 equ b'0101' ;Step index position 1
Index2 equ b'0001' ;Step index position 2
Index3 equ b'1001' ;Step index position 3
Index4 equ b'1000' ;Step index position 4
Index5 equ b'1010' ;Step index position 5
Index6 equ b'0010' ;Step index position 6
Index7 equ b'0110' ;Step index position 7
Index8 equ b'0100' ;Step index position 8
Threshold1 equ b'10111101' ;Threshold level for minimum light detection
Threshold2 equ b'11101011' ;Threshold level for light search subroutine
org 0x000 ;Reset vector
nop
```

```
;**Initial setup**
Initial
banksel PORTC ;Select Bank 0

clrf PORTC ;Clear PORTC
movlw B'01000001' ;Configure ADCON0
movwf ADCON0 ;ADCO0 = Fosc/8, Ch0, AD converter on
banksel OPTION_REG ;Select Bank 1
movlw B'10000110' ;Configure OPTION_REG
movwf OPTION_REG ;TMR0 prescaler = 1:128
clrf TRISC ;Set PORTC as all outputs
movlw B'00000110' ;Configure ADCON1
movwf ADCON1
banksel PORTC ;Select Bank 0
movlw Index1 ;Set initial step motor position at index position
movwf PORTC
clrf PositionCount ;Zero out position
;**Search for brightest point (find sun) following initial startup**
Search
btfss INTCON,T0IF ;Checks if Timer0 interrupt flag is set
goto Search ;Loops until set
bcf INTCON,T0IF ;Clears Timer0 interrupt flag
bsf ADCON0,GO ;Sets GO bit in ADCON0 to start ADC
Wait2
btfss PIR1,ADIF ;Checks if AD interrupt flag is set
goto Wait2 ;Loops until conversion is complete
bcf PIR1,ADIF ;Clears AD interrupt flag
movlw Threshold2 ;Moves minimum brightness value to W for comparison
subwf ADRESH,W
btfsc STATUS,C ;See if value in ADRESH is greater than threshold value
goto Main ;If ADRESH > min threshold, continue to main routine
movlw Right ;If ADRESH < min value, move motor right
movwf Direction
call StepControl
incf PositionCount,1 ;Increment position count for right movement
call PositionCountCheck ;Check location to see if at end of day
goto Search ;Keep looking for brightest point (sun)
Delay
;call Delay1s ;4 sec delay for class demo
;call Delay1s
;call Delay1s
;call Delay1s
call Delay1m ;4 minute delay for actual implementation
call Delay1m
call Delay1m
call Delay1m
;**Main move/measure/compare routine**
Main
call ADCStart ;Call ADC to get measurement at center position
```

```
movf ADRESH,W

movwf Position0 ;Store center position measurement
movlw Left ;Move motor left 1 step
movwf Direction
call StepControl
decf PositionCount,1 ;Decrement position count for left movement
call PositionCountCheck ;Check location to see if at end of day
call ADCStart ;Call ADC subroutine to get measurement at left position
movf ADRESH,W
movwf Position1 ;Store left position measurement
movlw Right ;Move motor right 2 steps
movwf Direction
call StepControl
incf PositionCount,1 ;Increment position count for right movement
call PositionCountCheck ;Check location to see if at end of day
movlw Right
movwf Direction
call StepControl
incf PositionCount,1 ;Increment position count for right movement
call PositionCountCheck ;Check location to see if at end of day
call ADCStart ;Call ADC to get measurement at right position
movf ADRESH,W
movwf Position2 ;Store right position measurement
movf Position0,W ;Check if center and left positions are equal
subwf Position1,W
btfsc STATUS,Z
goto ReturnToPosition0 ;If equal, return to center position
movf Position0,W ;Check if center and right positions are equal
subwf Position2,W
btfsc STATUS,Z
goto ReturnToPosition0 ;If equal, return to center position
movf Position1,W ;Check if left and right positions are equal
subwf Position2,W
btfsc STATUS,Z
goto ReturnToPosition0 ;If equal, return to center position
movf Position1,W ;Check if center position is greater than left position
subwf Position0,W
btfsc STATUS,C
goto NextCheck ;If center is greater than left, compare to right position
goto NextMajorCheck ;If center is less than left, check left position vs. others
NextCheck
movf Position2,W ;Check if center position is greater than right position
subwf Position0,W
btfsc STATUS,C
goto ReturnToPosition0 ;If center position is greatest, return to center position

goto NextMajorCheck ;If center position is not greatest, check left vs. others
ReturnToPosition0
```

```
movlw Left ;Moves motor one step left to center if center is greatest
movwf Direction
call StepControl
decf PositionCount,1 ;Decrement position count for left movement
call PositionCountCheck ;Check location to see if at end of day
goto Delay
NextMajorCheck
movf Position0,W ;Check if left position is greater than center position
subwf Position1,W
btfsc STATUS,C
goto NextCheck2 ;If left position is greatest, compare to right position
goto NextMajorCheck2 ;If left is less than center, check right position vs. others
NextCheck2
movf Position2,W ;Check if left position is greater than right position
subwf Position1,W
btfsc STATUS,C
goto ReturnToPosition1 ;If left position is greatest, return to left position
goto NextMajorCheck2 ;If center in not greatest, check right position vs. others
ReturnToPosition1
movlw Left ;Moves motor two steps left to left if left is the greatest
movwf Direction
call StepControl
decf PositionCount,1 ;Decrement position count for left movement
call PositionCountCheck ;Check location to see if at end of day
movlw Left
movwf Direction
call StepControl
decf PositionCount,1 ;Decrement position count for left movement
call PositionCountCheck ;Check location to see if at end of day
goto Delay
NextMajorCheck2
movf Position0,W ;Check if right position is greater than center position
subwf Position2,W
btfsc STATUS,C
goto NextCheck3 ;If right position is greatest, compare to left position
goto ReturnToPosition0 ;If right is less than center, return to center position
NextCheck3
movf Position1,W ;Check if right position is greater than left position
subwf Position2,W
btfsc STATUS,C
goto DoNothing ;If right position is greatest, remain in place
goto Delay ;If not, start over

DoNothing
goto Delay
; **Step motor control**
ADCStart
btfss INTCON,T0IF ;Checks if Timer0 interrupt flag is set
goto ADCStart ;Loops until set
```

```
bcf INTCON,T0IF ;Clears Timer0 interrupt flag
bsf ADCON0,GO ;Sets GO bit in ADCON0 to start ADC
Wait
btfss PIR1,ADIF ;Checks if AD interrupt flag is set
goto Wait ;Loops until conversion is complete
bcf PIR1,ADIF ;Clears AD interrupt flag
movlw Threshold1 ;Moves min light intensity value to W for comparison
subwf ADRESH,W
btfsc STATUS,C ;See if value in ADRESH is greater than threshold value
return ;If ADRESH > min threshold, continue w/ main routine
goto ADCStart ;If ADRESH < min value, keep checking
StepControl
call Delay1s ;Delay between step movements
movf Direction,W ;Find out direction to move
movf PORTC,W ;Read PORTC
sublw Index1 ;Compare PORTC to index position
btfss STATUS,Z ;If no match, check another against another position
goto StepControl2
movf Direction,W ;Find out direction to move
sublw Right ;See if right is the direction called for
btfsc STATUS,Z ;If right, proceed to next position in the right direction
goto StepControl1
movlw Index2 ;If not right, proceed to next position in the left direction
goto MoveMotor
StepControl1
movlw Index8 ;Next index position in the right direction
goto MoveMotor
StepControl2
movf PORTC,W ;Read PORTC
sublw Index8 ;Compare PORTC to index position
btfss STATUS,Z ;If no match, check another against another position
goto StepControl4
movf Direction,W ;Find out direction to move
sublw Right ;See if right is the direction called for
btfsc STATUS,Z ;If right, proceed to next position in the right direction
goto StepControl3
movlw Index1 ;If not right, proceed to next position in the left direction
goto MoveMotor
StepControl3

movlw Index7
goto MoveMotor ;Next index position in the right direction
StepControl4
movf PORTC,W ;Read PORTC
sublw Index7 ;Compare PORTC to index position
btfss STATUS,Z ;If no match, check another against another position
goto StepControl6
movf Direction,W ;Find out direction to move
sublw Right ;See if right is the direction called for
```

```
btfsc STATUS,Z ;If right, proceed to next position in the right direction
goto StepControl5
movlw Index8 ;If not right, proceed to next position in the left direction
goto MoveMotor
StepControl5
movlw Index6 ;Next index position in the right direction
goto MoveMotor
StepControl6
movf PORTC,W ;Read PORTC
sublw Index6 ;Compare PORTC to index position
btfss STATUS,Z ;If no match, check another against another position
goto StepControl8
movf Direction,W ;Find out direction to move
sublw Right ;See if right is the direction called for
btfsc STATUS,Z ;If right, proceed to next position in the right direction
goto StepControl7
movlw Index7 ;If not right, proceed to next position in the left direction
goto MoveMotor
StepControl7
movlw Index5 ;Next index position in the right direction
goto MoveMotor
StepControl8
movf PORTC,W ;Read PORTC
sublw Index5 ;Compare PORTC to index position
btfss STATUS,Z ;If no match, check another against another position
goto StepControl10
movf Direction,W ;Find out direction to move
sublw Right ;See if right is the direction called for
btfsc STATUS,Z ;If right, proceed to next position in the right direction
goto StepControl9
movlw Index6 ;If not right, proceed to next position in the left direction
goto MoveMotor
StepControl9
movlw Index4 ;Next index position in the right direction
goto MoveMotor
```

```
StepControl10
movf PORTC,W ;Read PORTC
sublw Index4 ;Compare PORTC to index position
btfss STATUS,Z ;If no match, check another against another position
goto StepControl12
movf Direction,W ;Find out direction to move
sublw Right ;See if right is the direction called for
btfsc STATUS,Z ;If right, proceed to next position in the right direction
goto StepControl11
movlw Index5 ;If not right, proceed to next position in the left direction
goto MoveMotor
StepControl11
movlw Index3 ;Next index position in the right direction
```



```
goto MoveMotor
StepControl12
movf PORTC,W ;Read PORTC
sublw Index3 ;Compare PORTC to index position
btfss STATUS,Z ;If no match, check another against another position
goto StepControl14
movf Direction,W ;Find out direction to move
sublw Right ;See if right is the direction called for
btfsc STATUS,Z ;If right, proceed to next position in the right direction
goto StepControl13
movlw Index4 ;If not right, proceed to next position in the left direction
goto MoveMotor
StepControl13
movlw Index2 ;Next index position in the right direction
goto MoveMotor
StepControl14
movf PORTC,W ;Read PORTC
sublw Index2 ;Compare PORTC to index position
btfss STATUS,Z ;If no match, check another against another position
goto StepControl16
movf Direction,W ;Find out direction to move
sublw Right ;See if right is the direction called for
btfsc STATUS,Z ;If right, proceed to next position in the right direction
goto StepControl15
movlw Index3 ;If not right, proceed to next position in the left direction
goto MoveMotor
StepControl15
movlw Index1 ;Next index position in the right direction
goto MoveMotor
StepControl16 ;Otherwise return to index position 1
movlw Index1
```

```
MoveMotor
movwf PORTC ;Move motor to desired index position
return
; **End of day checking/reset/sleep routine**
PositionCountCheck
movf PositionCount,W
sublw d'48'
btfsc STATUS,Z ;If equal to 48, go to end of day brightness check,
otherwise, see if equal to 208
goto EndOfDayBrightCheck
movf PositionCount,W
sublw d'208'
btfsc STATUS,Z ;If equal to 208, go to end of day brightness check,
otherwise, return to main routine
goto EndOfDayBrightCheck
return
EndOfDayBrightCheck
```

```
btfs INTCON,T0IF ;Checks if Timer0 interrupt flag is set
goto EndOfDayBrightCheck ;Loops until set
bcf INTCON,T0IF ;Clears Timer0 interrupt flag
bsf ADCON0,GO ;Sets GO bit in ADCON0 to start ADC
Wait3
btfs PIR1,ADIF ;Checks if AD interrupt flag is set
goto Wait3 ;Loops until conversion is complete
bcf PIR1,ADIF ;Clears AD interrupt flag
movlw Threshold1 ;Moves minimum light intensity value to W for
comparison w/ ADRESH
subwf ADRESH,W
btfs STATUS,C ;See if value in ADRESH is greater than threshold value
goto EndOfDayBrightCheck ;If ADRESH > minimum threshold, continue checking
call ReturnToEast ;Otherwise, return to east and go to sleep
sleep
goto Initial
ReturnToEast
movlw d'48' ;Repeat 48 times to return motor to start (east)
movwf Temp
Loop
movlw Left ;Move motor left 1 step
movwf Direction
call StepControl
decfsz Temp,F
goto Loop
return
; **Delay routines**
Delay1ms
movlw d'250' ;1 msec delay
movwf Time1
25
Loop1
nop
decfsz Time1,F
goto Loop1
return
Delay250ms ;250 msec delay
movlw d'250'
movwf Time2
Loop2
call Delay1ms
decfsz Time2,F
goto Loop2
return
Delay1s ;1 sec delay
movlw d'4'
movwf Time3
Loop3
call Delay250ms
```

```
decfsz Time3,F
goto Loop3
return
Delay1m ;1 min delay
movlw d'240'
movwf Time4
Loop4
call Delay250ms
decfsz Time4,F
goto Loop4
return
end
```