

三等奖

# 以FPGA实现智能型太阳能追踪控制系统

大学院校： 元智大学/ 电机工程研究所

参赛队员： 张欣宏 吴宗宪 余政达

指导教师： 黄英哲

## 一. 设计概述

### 1. 设计背景

由于现今高科技环境下，能源是促进经济发达和社会进步的原动力。从工业革命以来，人类所使用的主要能源为石化能源，然而其蕴藏量有限，大量使用造成全球环境生态和气候产生莫大的变化，同时大气中的温室气体浓度大幅提高，造成气温逐渐升高、海平面上升等温室效应的现象，威胁了我们生存的环境。因此在环保意识抬头的今日，积极开发低污染及低危险性能源乃为迫切的需要。而太阳能是一种免费、无污染且又取之不尽、用之不竭，是一种可以利用来发电的一种相当好的能源。而目前的太阳能发电效率普遍都不高，所以如何增加太阳能发电系统的发电效率是值得研究的。

由于太阳会随著四季的季节和天气有规律的变化，而太阳能板如能一年里都和太阳成垂直时，其接收日照强度是最好的。目前太阳能电池板的架设大都采固定式，其电池板的板面固定朝向天空中某一方向，并不随著太阳移动而偏向。要增加太阳光照射于太阳能电池板上的单位面积照度，有鉴于此，本研究设计追日性能良好的太阳追踪发电系统。而所设计出的机构，除了必须能承载太阳能电池面板，也要能让面板在空间中做接近半球面的立体转动，如此才能完全追踪白天太阳在天空中的位置，也因此可以使总体发电量提高。此系统可以提高

照射能量密度，取得光照的最大量、以及在相同的发电量下，使用较少的太阳能电池以降低发电成本，因此极具研究发展的重要性。

## 2. 太阳能追踪控制系统

吾人所设计出的系统架构，使用两颗电机做为驱动源来设计出追日性能良好的太阳追踪装置器，此机构接近半球面立体转动的数组太阳能发电系统，在一样维持两个驱动源(电机)的条件下，要具备有可以组成数组型式的功能，如图1 所示。它的优点是仅用两个驱动电机，并且没有耦合上的问题，也就是其中一颗电机的转动角度并不会影响另一颗电机所需要转动的角度，这可以有效降低控制上的困难度。并且此一追踪器并无一般常见双轴运动机构中，有一颗电机必须承载另一颗电机重量的缺点，如此一来，系统的运作能量消耗可以降至最低，使总体发电量提高，也使太阳能发电系统更为经济。

基于以上的考虑，且在机构必须能承载太阳能电池面板，并使得面板能在空间中做某一范围的立体转动的前提下，太阳追踪器如图2 所示。

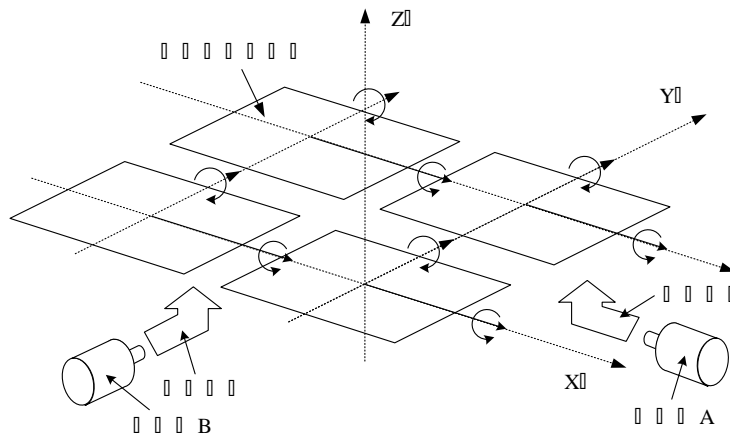


图1 数组太阳追踪架构示意图

本作品之此数组式机构有两个最重要的优点：(1) 光电转换效率高。由于数组太阳追踪器的活动面板具有立体转动的功能，且数组太阳追踪器可以定时追日，因此，系统具有很高的光电转换效率，拥有多产能的优点；(2) 机构控制简洁、省能源。数组太阳追踪器追日的两个转动维度，由两个转动角度不相耦合且无需要承载另一个驱动源重量的独立驱动源所控制，同时，旋动面板的整体转动惯量也大幅降低，因此，可以具有系统受控机制简洁且节省运作能源的双重优点。

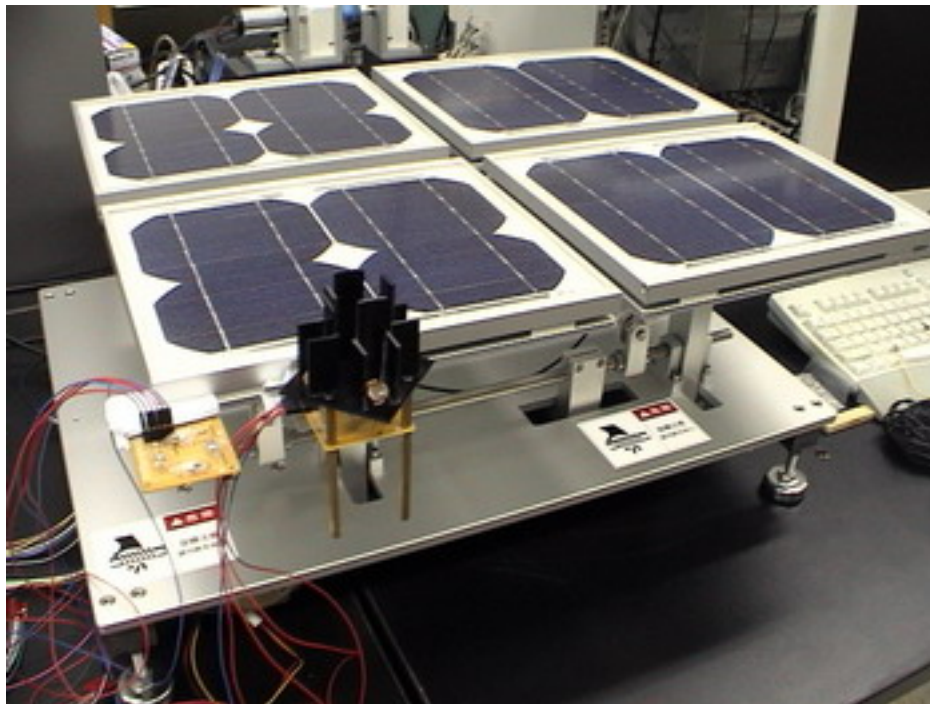


图2 完整太阳能追踪控制平台

本作品主要以ALTERA FPGA内含一个 Nios微处理器来结合双轴电机追踪控制器，其中以SOPC(System-On-a-Programmable-Chip)概念设计，将微处理器、内存、I/O端口等外围器件集成到一个芯片上，这样集成的优点：加速开发时程，并兼顾设计弹性；采用单芯片的解决方案，电路板可以缩小，节省相关成本；产品测试将变得较为容易，基于上述本系统，以Altera Nios II 完成太阳能追踪控制。

## 二. 功能描述

在此我们主要设计分成三种模式：

### 1. 平衡定位

在太阳能平台上初始值时候，利用水银开关做为我们的平衡定位，目的是可以设定四周边界问题，以防止因为边界过于倾斜而平台上四面太阳能板子撞至机构平台，造成电机及平台的损坏。

### 2. 自动模式

以光敏电阻的受光大小作为我们光源判断为主要追光传感器，可利用此传感器依据太阳光大小经由模拟转数字信号IC回授至FPGA控制器，以Nios II微处理器作为主要控制核心，调整双轴电机将平台调至最佳发电效率位置。

### 3. 手动模式

当系统发生故障、保养等，我们可以将系统调至手动模式，任意调整位置来进行人员检修。

整体太阳追踪装置系统的方框图，如图3 所示。

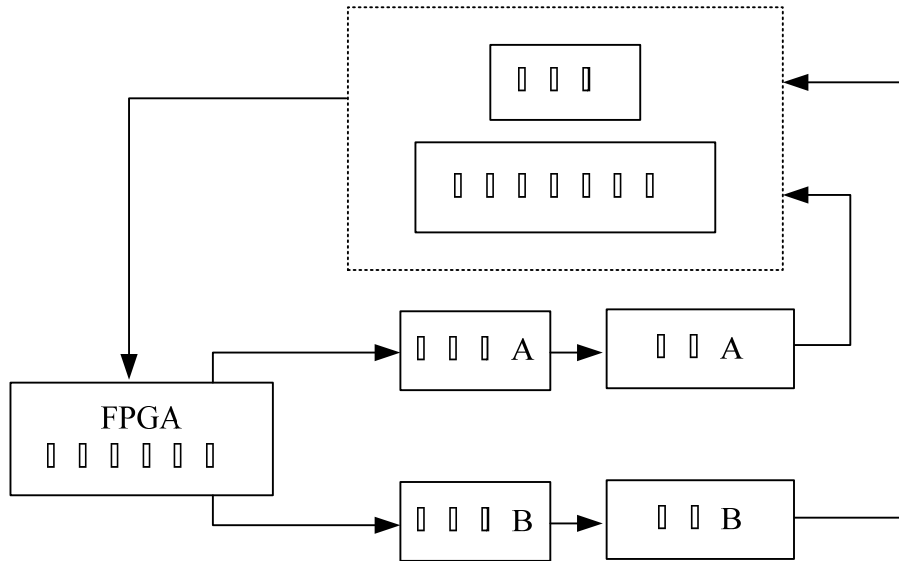


图3 太阳追踪器服务控制方框图

本系统追踪的逻辑流程设计是通过Nios微处理器所构成的控制电路来实现。追踪控制的主要流程如图4所示，其中，FPGA程序流程中的”开始”定义为启动追控电路的电源开关，配合所设计的控制模式和防止外部干扰措施，来达到系统追踪、省能以及保护的的目的。其中外部干扰，包含自然天候的影响，例如，风、沙、雨雪、冰雹及盐害（盐对机械机构等的侵蚀）等等。

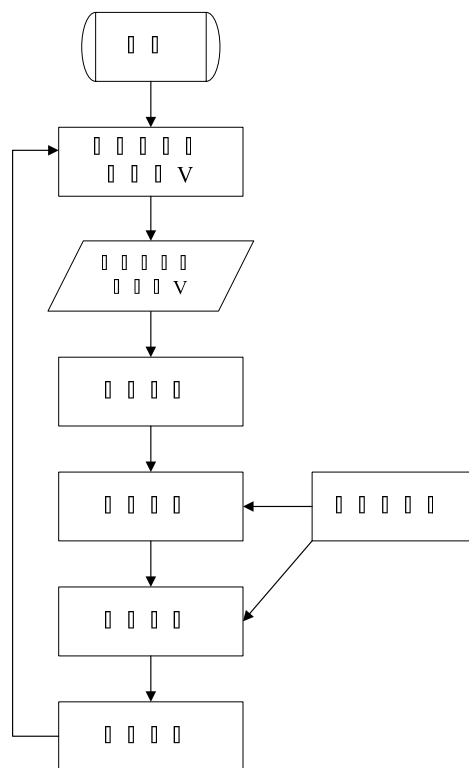


图4 FPGA程序流程图

### 三. 性能参数

在这里我们使用了以下模块来组成我们的太阳能追踪系统。

- (1) 平衡传感器模块：在系统起始复归时，四个开关皆导通为我们平衡水平状态。
- (2) 模拟/数字模块：对应 A/D 电子零件为 AD0804，其转换时间为  $100 \mu s$ ，为 8 比特分辨率的 IC。
- (3) 电机控制模块：使用步进电机  $100 \mu s$  周期来控制电机速度。
- (4) 中央处理器：自定制的 32 比特基于 RISC 结构 CPU，要求实现对太阳能系统的控制。

在要求精密度上，在模糊控制上时间为不超过 0.1 秒。平衡传感器模块目的除了复归平衡外，定位水平为  $0^\circ = 262144$ ，另一功能为 X 和 Y 轴的边界范围。

## 四. 设计结构

以图5 所示，我们将以Nios II微处理器为我们的控制核心，并集成我们设计的两轴控制芯片结合。通过光传感器所传回授数据至FPGA内判断，进行追踪控制法则的演算，如此便可以计算出电机所需要角度，并能做出适当的调整，进而移动太阳能板达到最佳获取功率之倾角。

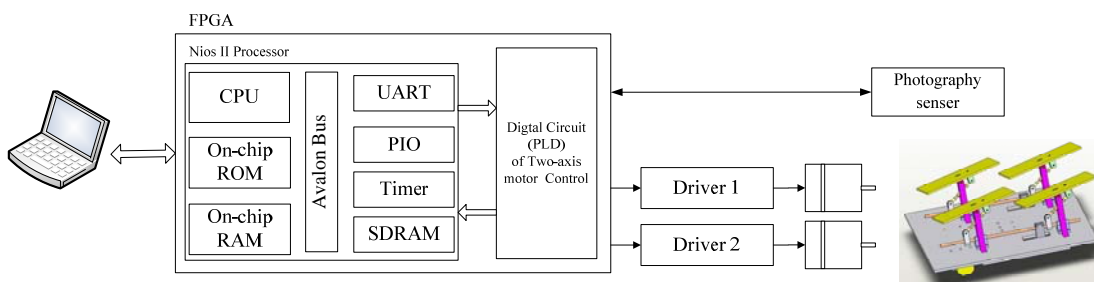


图5 系统结构图

### 1. 硬件设计

首先我们先通过平衡传感器作为系统开始的归零动作，然后将我们设计一传感器用以追踪传感器来判定太阳光源的方位所在，依据感测传回信号作为判断依据成为我们控制器的输入，经由控制设计，输出信号来控制双轴步进电机，进而达到我们太阳能追踪控制系统，以下分别说明硬件介绍。

#### (1) 平衡传感器

在开始归位平衡时，采用水银开关，又称倾侧开关，是电路开关的一种，以一接著电极的小巧容器存储著一小滴水银，容器中多数注入惰性气体或真空，如图6a 所示。因为重力的关系，水银水珠会随容器中较低的地方流去，如果同时接触到两个电极的话，开关会将电路闭合，开启导通开关。依据此特性在机构设计中分别于东西及南北方向加入四个开关，并固定装置，如果四个开关皆导通则此机构保持平衡，实体图如图6b 所示。



图6a 水银开关



图6b 平衡传感器实体图

## (2) 传感器设计

本研究的设计重点之一是传感器的设计，用以追踪传感器来判定太阳光源的方位所在，故搜寻传感器选用非常重要。由光敏电阻(Cadmium Sulphide, 简称为Cds), 如图7a 所示, 不但成本便宜, 在设计一个具高可靠度且感光功能良好的传感器, 具有下列良好的特性, 必须先了解传感器器件的特性, 以下将说明传感器的说明。

- (1) 光敏电阻是属于无极性(欧姆结构), 故使用上简易方便。
- (2) 光敏电阻会随著光能强度使内阻抗改变的光可变电阻。
- (3) 光敏电阻于周围环境光变亮时, 内电阻会下降。
- (4) 光敏电阻对于光的敏感性(即光谱特性)约在 0.4~0.8  $\mu\text{m}$  和太阳光的可见光范围波长 0.38~0.76  $\mu\text{m}$  的响应很接近, 如图 7b 所示。

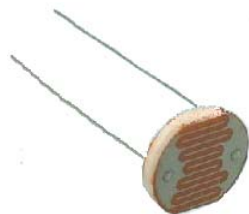


图7a 光敏电阻实体图

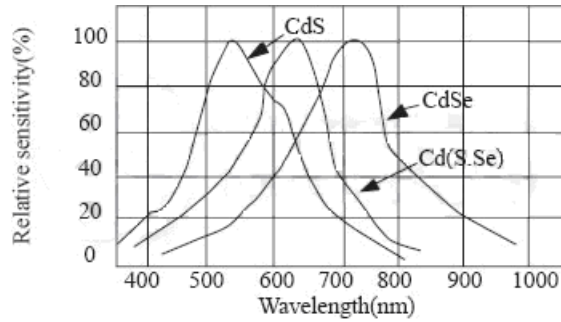


图7b 光敏电阻敏感度范围

### (3) 追踪传感器设计

追踪传感器主要经由四颗特性相近光敏电阻构成，负责侦测东、西、南、北四个方向的光源强度，于各方向均有一个光敏电阻，并以45度角朝向光源处，并将该方向设置基座以将该方向以外的光线隔离，以达到快速判别太阳位置的广角式搜索，如图8a所示。四个传感器分为两组，一组是两个光敏电阻做为东西向的传感器，用以比较东西向受光强度的差异。当东西向的传感器接受到的光源强度不一致时，系统会依据东西向两传感器输出电压得到信号，使用电压型模拟数字转换(A/D Converter)、IC(ADC0804)判读到的传感器输出电压数值的不同，判断该方向的受光较强，并且驱动步进电机朝向该方前进，当东西向传感器输出值相等时，则输出的差值为零、电机驱动电压亦为零，即追踪到太阳目前的位置。另一组的南北向传感器，则是相似的原理，用来追踪太阳在南北向的位置，追踪传感器实体外观图如图8b所示。

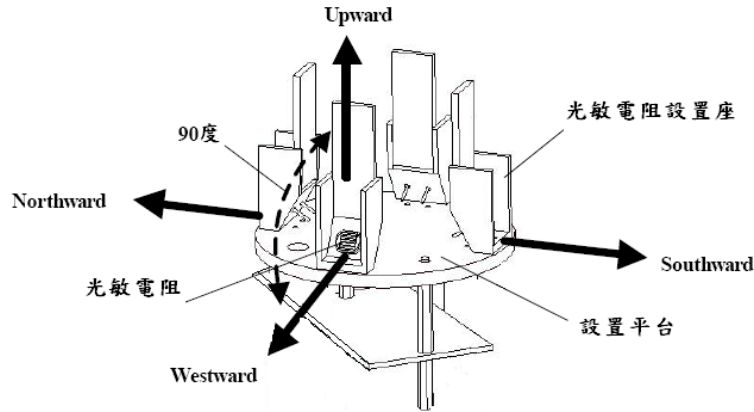


图8a 追踪传感器内部设计图

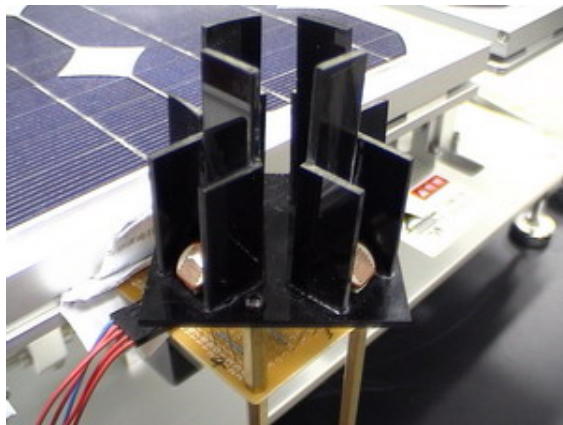


图8b 追踪传感器实体外观图

#### (4) A/D 转换器

一般所量测的电压或电流等连续信号，可称之为模拟信号。把模拟信号转换成数字信号的装置则称为模拟对数字转换器(analog to digital converter, ADC)。数字信号在传送中即使有噪声干扰，它仍然可以把噪声所造成的干扰降至最低，且可利用编码技术使其具有调试的功能；除此之外，数字信号尚有容易存储的优点。

在ADC 转换IC 的选择上，ADC0804 为一具有8比特分辨率单信道的20只引脚IC，其使用5V 单电源供应工作。模拟输入电压范围为0V至5V，消耗功率为15mW，转换时间为100  $\mu$ s。由于具有8比特分辨率，故有256阶的量化，若参考电压为5V，则每一步阶为 $5/256=0.01953$ V。00000000(00H)表示0.00V，11111111(FFH)表4.9805V。ADC0804 的不可调误差(unadjusted error)为 1 LSB，即0.01953V，其中包含满刻度误差、补偿(offset)误差及非线性误差。

图9a 为ADC0804 之引脚图，其中 $D_0$ 到 $D_7$ 为8比特输出埠。当 $\overline{CS}$  (第1脚)与 $\overline{RD}$  (第2脚)皆为低位准(low)时，数字数据则会被送到输出埠。若 $\overline{CS}$ 与 $\overline{RD}$ 任一引脚为高位准(high)时，则 $D_0$ 到 $D_7$ 为浮接状态。 $\overline{WR}$  (第3脚)可视为启动转换的控制信号，当 $\overline{CS}$ 与 $\overline{WR}$ 皆为低位准时，ADC0804 做清除的动作，当 $\overline{WR}$ 回到高位准时，ADC 便会开始转换。 $CLK IN$ (第4脚)为时序输入，频率范围为100 KHz 至800 KHz。 $\overline{INTR}$  (第5脚)在转换期间为高位准，直到转换完毕时 $\overline{INTR}$ 会变为低位准。第6脚 $V_{in}(+)$ 及第7脚 $V_{in}(-)$ 为差动模拟信号输入，通常使用单端输入，而将 $V_{in}(-)$ 接地。ADC0804 有两个接地端，一为第8脚模拟接地(A GND)及第10脚数位接地(D GND)。第9脚( $V_{ref}/2$ )是二分之一参考电压输入值，若空接则 $2 V_{ref}$ 等于电源电压 $V_{CC}$ 。ADC0804 内建史密特触发器(Schmitt trigger)，如图9b 所示。若在第19脚 $CLK R$ 及第4脚 $CLK IN$ 加上电阻及电容，则可生成ADC 工作时所需的时序，其频率约为

$$f_{CLK} \approx \frac{1}{1.1RC} \text{ (Hz)} \quad (1)$$

以此方式便可由R 和C 来决定时序信号，可不用再由CLK IN 端外加时序信号。图9c 为ADC0804 模拟对数字转换器电路，输入的模拟信号大小由可变电阻VR<sub>2</sub> 控制并由V<sub>in</sub> (+) 端输入，V<sub>in</sub> (-) 短路；2 V<sub>ref</sub> 由R<sub>1</sub>、R<sub>2</sub> 及VR<sub>1</sub> 提供；C<sub>1</sub> 及R<sub>3</sub> 用来控制电路的时序，CS 与RD 两者均接地以使芯片致能(enable)，WR 与INTR 接到SW<sub>1</sub> 开关以便仿真控制信号。

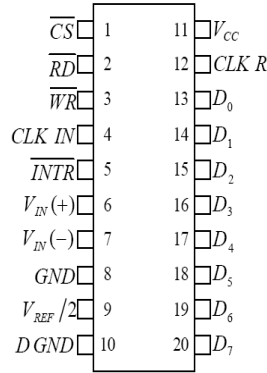


图9a ADC0804 引脚功能图

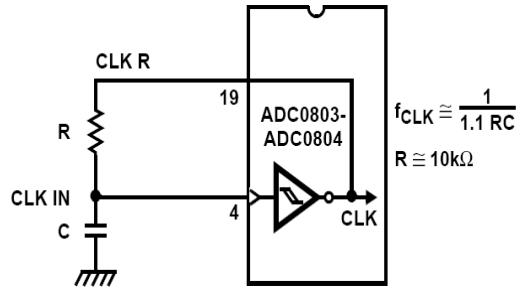


图9b ADC0804 内部时序生成电路

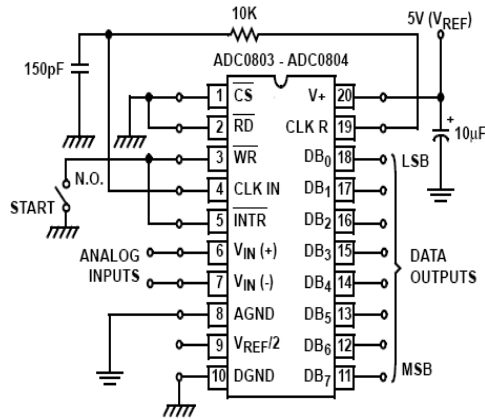


图9c ADC0804 模拟数字转换器电路图

表1 ADC 详细规格表

工作电压	+5V DC
模拟电压输入范围	$0 \leq V_{in} \leq +5V$ DC
分辨率	1/256
转换输出值	0 ~ 255
转换频率	$f_{ck} = 1 / (1.1 \times R \times C)$
转换误差	$\pm 1$ LSB
参考电压	+2.5V DC

ADC转换器完整电路实体图如图9d 所示。

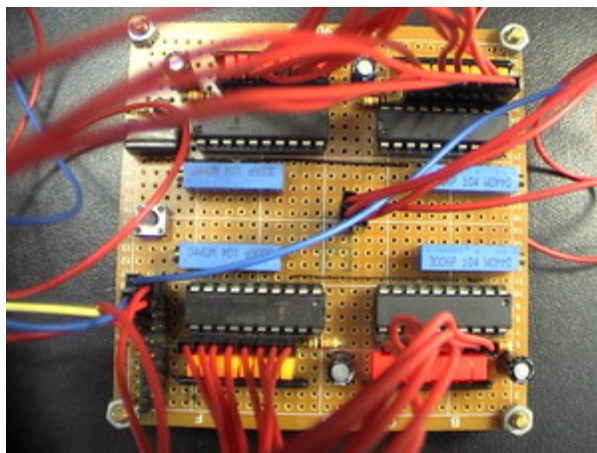


图9d ADC转换器完整电路实体图

(5) 实体传感器制作

依照前述所介绍，水银开关制作成初始值平衡传感器实体电路接线图如图10a 所示。

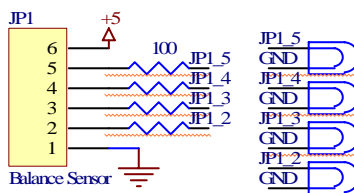


图10a 平衡传感器实体电路接线图

依据太阳光源所感测的光敏电阻，其输出信号经由ADC芯片第6脚作为输入信号，转换成模拟信号由第11脚至18脚为8比特输出信号，之后接至我们FPGA上为输入信号判断，其完整的电路接线图如图10b 所示。

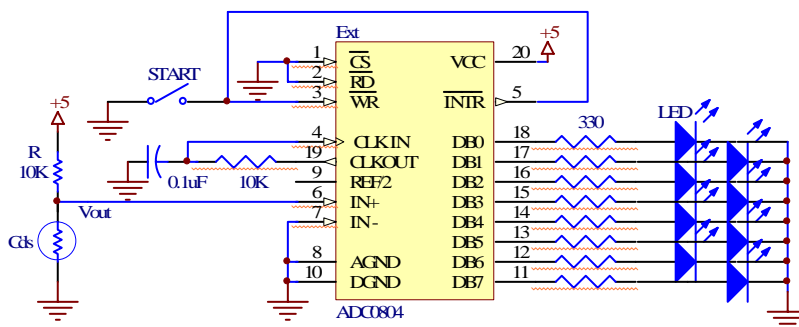


图10b ADC实体电路接线图

## 五. 设计方法

### 1. 模糊逻辑控制设计

#### (1) 模糊理论简介

自从1965年美国加州柏克莱大学的L. A. Zadeh教授在“信息和控制”(Information and Control)学术杂志上,提出了模糊集合(fuzzy sets)的概念,模糊理论(fuzzy theory)就开始蓬勃发展,因为模糊理论强调人类的知识大部分都可以用语言来表达,也就是可以将所有的知识领域加以模糊化,它的特色较偏重于人类的经验及对问题特性的掌握程度,而不主张用繁杂的数学分析及数学模式来解决问题。模糊化的好处是可以提供更佳的推广性、错误容忍性、及更适合应用在真实世界的非线性系统。

早在1974年英国伦敦大学Queen Mary分校的E. H. Mamdani教授,(fuzzy inference),成功地将模糊控制应用在蒸汽机自动运转上。而1980年由丹麦Smidth公司所开发完成水泥窑炉的模糊控制,将水泥窑炉运转时的各种状况及其对策,转换成语言式的控制规则,再借著微电脑系统的操作来达到控制的目的。尤其是在1987年7月开始营运的日本仙台市地铁,由于使用了模糊控制,使得列车的自动运转系统成功地走上实用阶段,也是一个非常成功的案例。此外一些家电产品也广泛应用此项技术。

模糊理论其实是一门相当生活化的学问,由于使用语言变量描述,因此更具亲和力和一般性,人类日常生活所用的一些叙述即大多是模糊的,例如那水果“很甜”,车开的“很快”等等,其中“很甜”、“很快”,并不是一个明确值,而只是描述事件的程度,但是根据这些描述,人类却能很轻易的接收模糊理论发展至今近四十年,相关的研究成果颇受肯定,应用的研究已延伸各类科技领域,包括了:

- 控制工程:如智能型控制、汽车电子、日本仙台市地铁…等。
- 图像识别:如图像处理、语音辨识、信号处理…等。
- 家电产品:如洗衣机、电冰箱、空调…等。
- 其它:数据管理、教学评量、财经管理…等。

## (2) 模糊逻辑控制器结构

模糊逻辑控制器(Fuzzy Logic Controller; FLC)结构如图11所示,一般模糊逻辑控制器分成四个部分,分别是模糊化接口、推论引擎、知识库以及解模糊化接口,以下就对四个部分分别介绍:

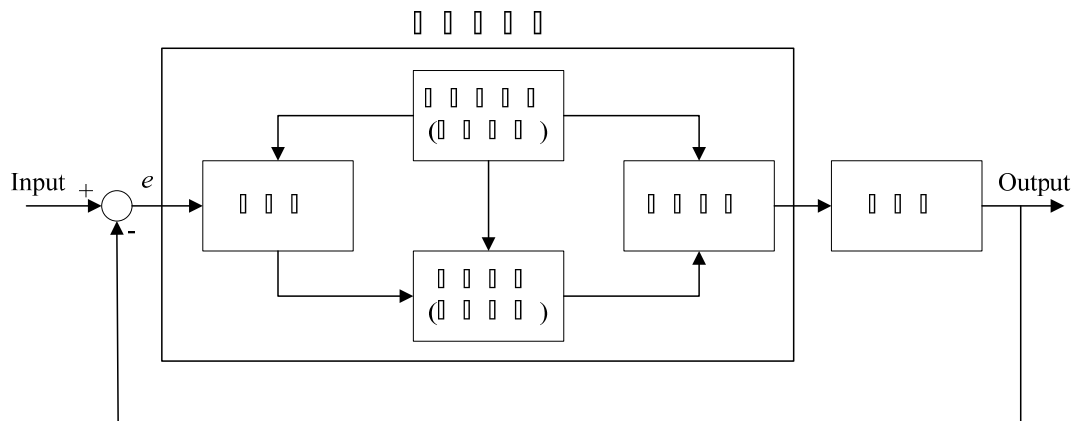


图11 模糊控制基本结构

### 1. 模糊化(Fuzzification)

一般控制器的输入通常为一个确定的数值,然而模糊控制的知识库是以语意形式所表达,将明确数值转为语意形式,将所观测量的数值大小转换到语意形式对等的论域中,以方便模糊推论机能进行推论动作,这样的转换过程称之为模糊化。

### 2. 知识库(Knowledge Base)

知识库是模糊控制的推论依据,用来定义所有使用到有关的语言控制法则及相关参数设定。整个模糊控制系统的数据核心就是以数据库为准,包含数据库(Data Base)和规则库(RulesBase)这两大部份:

#### (a) 数据库

模糊控制规则表达通常有两种形式,第一种方式为状态评价形式(State evaluation type),此种方式是利用时间 $t$ 去衡量系统状态,并以数据库的控制规则和输入变数来计算时间 $t$ 时的模糊控制动作,另外一种为目标评价形式(Object evaluation type),这种方式是先预估目前及未来的控制动作,并对控制目标是否达成所要求的评价,再决定加以输出控制命令。而数据库通常由以下来源所建立生成:

- 工作经验及专家经验法则:模糊控制的规则是依据受控系统的信息所得,因此经验法则是属最重要的部份。
- 系统自我学习:此方法利用离线学习的方式,由其它演算法辅助去建立规则数据库,但这个方法要先建立起系统的模型。
- 预测评估:这个方法可说是土法炼钢,要利用反复测试去验证结果,而最新所得的结果来修改数据库规则,但此方法需要较长的时间。

#### (b) 规则库

所谓规则库即包含许多以语言形式表现的规则，以下所表现出来的简单条件式叙述语句：

$$Rt : IF x \text{ is } A_t \text{ THEN } y \text{ is } B_t \quad (2)$$

其中  $t$  为叙述语句的编号，句中IF 部分称为叙述语句的前题(Antecedent)或称前件部，来提供条件以判断此语句成立与否，而THEN 部分则为叙述语句的结论(Consequent)或称后件部，用来表现判断符合条件的推论结果。前件部 $x$  是模糊系统的输入变数，用来观测系统状态，后件部 $y$  则是模糊系统中的输出变数，用来控制受控系统。实际变数量可依被控制系统的状况来作增减，而规则中的 $A_t$ 及 $B_t$ ；是利用语言形式所表现的模糊性概念，例如：高矮胖瘦的定义、速度快慢的定义等，因这些都存有人类主观意识，并不容易以数据来表示之，所以采用归属函数的方式定义。

### 3. 模糊推论机(Fuzzy Inference Engine)

模糊控制最重要的阶段在于模糊推论机，其主要工作于实际的决策过程处理，而它的基本理论就是近似推理，模糊推论机中有两种重要的推论方式，分别为GMP(Generalized modus ponens)及GMT(Generalized modus tollens)。其GMT 是目标驱动式的逆向模糊理论，然而GMP 为前向联结的推论方式，当进行数据输入时，便可按照规则将输出推论出，因此非常适合应用于模糊控制的推论机构上，而实际的动作由下面三个计算工作来完成：

- (a) 将被触动的规则前件部各个命题做AND 计算，以求得前题部的适合度。
- (b) 被触动规则前件部的适合度对应的结论部各个命题做AND 计算，以求得该规则成立的强度。
- (c) 最后完成所有被触动规则各结论部做OR 计算。

### 4. 解模糊化(Defuzzification)

解模糊化的过程刚好和模糊化相反，目的在于将模糊推论机的输出值，转换成相对等的明确值，使得这个明确值能符合受控系统的输入信号，此过程是便于将输出控制信号给予受控系统。

#### (3) 太阳能控制的制作

太阳追踪装置的两个驱动电机虽然在角度的转动上能独立运作，没有耦合上的困扰，但是在转动惯量的问题上，无可避免的，一定会有非线性的现象存在(这是立体转动机构共有的问题)。因此，电机的闭回路控制必然是需要的。在转动惯量的控制上，虽然有非线性的现象存在，但是太阳追踪装置所要求的转动速度因为太阳移动缓慢的关系并不需要太快。所以，模糊控制法则足够用来控制电机的运作，可以使得系统的控制机制有很大的可调变性和快速反应能力。

我们将采用模糊控制理论来做为电机驱动的控制基础。实现于硬件控制电路的实现，将使用硬件描述语言(VHDL、Verilog等)把控制程序加载Nios II微处理器中作为控制核心，再搭配传感器、译码器等，形成完整的控制回路，这样的安排可以使得系统发电量效能达到最佳。

#### (4) 模糊控制器的实现

本研究所设计控制器采用传感器接收光的强弱的测量值做为回授，经过一次次

的修正，最后达到控制目的，基本的模糊控制系统结构图如图12 所示。光敏电阻的电阻值随著光的强弱大小经由ADC转换所得到的分压电压值大小，由垂直(南北)及水平(东西)方向两轴的误差，模糊控制以这两组的误差分别作为模糊控制的输入。

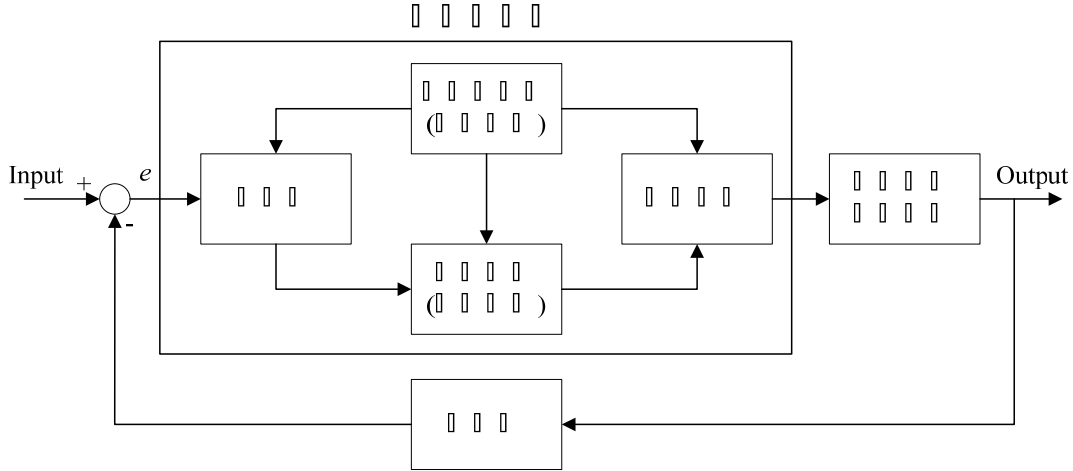


图12 太阳能模糊控制系统结构图

整个模糊控制器的设计，大致上可分为五个步骤来进行设计：

步骤1. 首先，先定义输入变数、输出变数及语意上的变数，其中各语意变量分别为：

输入变量的选择

本研究其中一轴选择输入变量为  $e$ ，而输出变量则为  $e$ ，定义所选择的1个语言变量分别为5个三角形归属函数。

输入变量

$$\text{误差 } e = b\_pio - a\_pio \quad (3)$$

输出变数

步进电机正转及反转调整之间秒数

输出变量的论域

采用正规化的离散论域，设定论域范围  $[-75, 75]$ 。

语言项的定义

一个语言变量都定义五个语言项，亦即

$$e = \{NB, NS, ZE, PS, PB\}$$

其中各语言项定义为

- NB: Negative Big (负大)
- NS: Negative Small (负小)
- ZE: Zero (零)
- PS: Positive Small (正小)
- PB: Positive Big (正大)

步骤2. 根据先前所定义的语言上变数，来建立输入归属函数，而(东西或南北)方向误差的归属函数图如图13 所示。

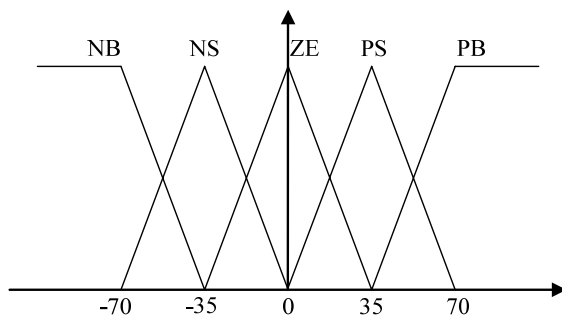


图13 模糊控制系统的归属函数

步骤3. 建立模糊规则库是最重要的一部份，因为所有的状态都必须根据我们所定义的规则库的规则进行操作，所以制定一个好的规则库是相当重要的，若要顺利构建一个模糊规则库，在此吾人采用5条模糊控制规则，此5条控制规则是以 " if...then..." 陈述的模糊控制规则来表示。

*Rule 1: If e is PB, then  $U_f$  is PB.*

*Rule 2: If e is PS, then  $U_f$  is PS.*

*Rule 3: If e is ZE, then  $U_f$  is ZE.*

*Rule 4: If e is NB, then  $U_f$  is NB.*

*Rule 5: If e is NS, then  $U_f$  is NS.*

步骤4. 定义其模糊推论引擎，模糊推论的方法有许多种，且不同的模糊推论会得到不一样的推论结果，而本文在此是藉由Mamdani所提出的重心法(center of gravity or center of area)，来作为解模糊化(Defuzzification)的工具，其主要因为这种方法较为简单而且可靠度高。

步骤5. 最后一个步骤，是将模糊推论引擎所推算出的结果，在经由解模糊化的过程，转换为精确的数值。而解模糊化也是有许多种方法，本文是采用解重心法来做解模糊化的过程，来达到实际的操作量。其公式如方程式4所示。

如果我们推论的结果为模糊单值，则权重式平均法是最为广泛使用的，假设有n条规则其中  $w_i$  为第  $i$  条模糊规则的启动强度， $r_i$  为第  $i$  条模糊规则推论的结果， $\hat{U}_f$  为输出操作量，则其表示式如下。

$$\hat{U}_f = \frac{\sum_{i=1}^n w_i r_i}{\sum_{i=1}^n w_i} \quad (4)$$

在这里我们实际以5条模糊规则，如下所示。

$$\hat{U}_f = \frac{\sum_{i=1}^5 w_i r_i}{\sum_{i=1}^5 w_i} = w_1 r_1 + w_2 r_2 + w_3 r_3 + w_4 r_4 + w_5 r_5 \quad (5)$$

这种解模糊化方法也最方便以数字电路的方式来实现。

## 2. FPGA芯片程序设计

近年来由于FPGA (Field Programmable Gate Array)的发展一日千里，工艺的进步让单一FPGA 芯片内可容纳的逻辑电路数量随之增加。将整个数字电子电路系统构建于单一芯片上，不管是速度还是电源消耗方面，都拥有相当大的优势，因此SOPC(System On a Programmable Chip)也渐渐成为一个设计的新趋势。SOPC 代表一种新的系统设计技术，可以将硬件系统（包括微处理器、内存、外设接口电路及用户逻辑电路）以及软件设计，都放至单一可编程芯片中。

## 3. SOPC系统设计

SOPC 系统发展流程如图14 所示，第一步骤是定义系统。这包括定义处理器、内存接口、外设器件、仲裁器(Arbitrator)以及客户指令(Custom Instructions)等方面。定义完系统后，进入系统生成(Generate System)阶段。这个阶段可以使用SOPC Builder 来完成这项工作。系统生成之后，分成两个方面进行设计，一是硬件设计，另外一个软件设计，在软件设计时，使用Quartus II 软件对硬件描述语言原始程序及EDIF 文件进行逻辑电路合成的编译，在软件设计时，使用GNUPro 软件开发工具和软件资源(如Header 文档、链接库、监督程序和外设驱动程序等)来生成和编辑应用程序代码，并配合使用Debug/Profile 进行程序调试的工作。为了确保软硬件设计正确性，还可利用ModelSim 软件进行仿真，一旦发现错误，便可回到系统生成阶段SOPC Builder 修改系统并生成系统，直到正确无误为止，再将硬件设计和软件设计依序下载至发展电路板或原型电路板做电路验证。

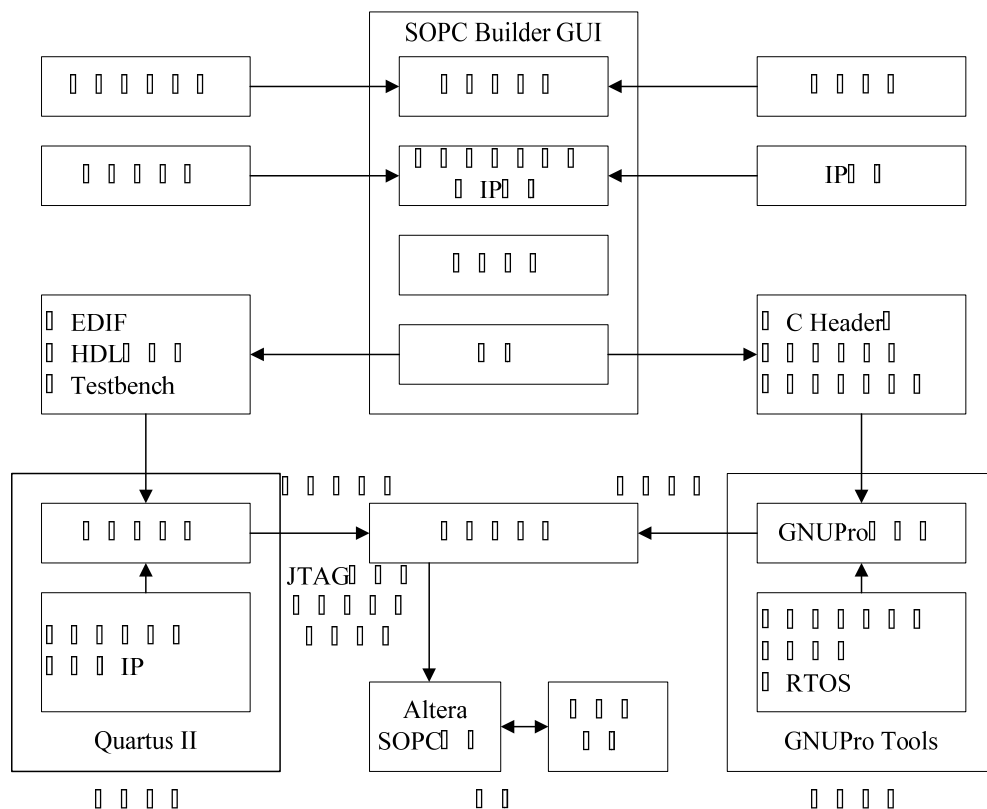


图14 SOPC 系统发展流程

### (1) 硬件设计解决方案

Altera 在嵌入式处理器方面，提供有Nios(Soft Core)和ARM(Hard Core)两种方式。Nios 是一种16/32 比特基于RISC 结构的可配置(Configurable)嵌入式处理器。在外围器件方面，提供有片内ROM、片内RAM、内存接口(如：SDRAM、SSRAM 和DMA 控制器)、串行I/O(如：UART 和Ethernet等)、并行I/O(如：输入/输出/双向端口或PCI 接口)以及定时器(如：简单定时器、频率定时器和看门狗定时器等)。在IP 方面，提供PCI 32/33桥接器和以太网的MAC。在总线方面，提供有Avalon 总线。

在EDA 硬件发展工具方面，Altera Quartus II(Place & Route 的工具)是设计输入(Design Entry)、编译及编程(Programming) SOPC 芯片的工具。Quartus II提供VHDL、Verilog、AHDL 或Block/Schematic Entry 等设计输入方式。Leonardospectrum做为电路综合工具。至于系统仿真方面则可使用ModelSim。

### (2) 软件设计解决方案

Altera 提供下列相关软件开发工具：

- 编译器 (Compiler)。
- 组译器 (Assembler)。
- 链接器 (Linker)。

调试器 (Debugger)。  
 监督程序 (Monitor)。  
 链接库和共用程序 (Libraries & Utilities)。

#### 4. Nios II嵌入式处理器

##### (1) Nios II 处理器系统简介

嵌入式系统最初为了工业计算机而设计，在信息产品普及和数字家电兴起后，嵌入式系统已逐渐普及到人们日常生活中，从信息家电、网路产品到可携式装置，都看得到它的踪迹。为了提高系统设计的集成程度，由于ARM微处理器并没有提供FPGA形式IP(Intellectual Property)，因而Altera公司以在FPGA内部嵌入ARM922T微处理器硬件电路，本研究中集成FPGA控制芯片仍以Nios以SOPC System On Programmable Chip)方式来发展嵌入式系统是一个方便可行的途径。以Altera公司之SOPC Builder集成设计环境为例，可以规划CPU的种类，例如ARM或Nios IP(以VHDL或Verilog语言软件形式嵌入CPU)为首选。SOPC Builder集成设计环境的主要优点是将所需电路及周遭器件集成刻录至一颗芯片，具有电路体积小以及软硬件都可以扩充、维护的优点，并且可以加速产品开发时程，兼顾设计弹性。

在此我们选用Altera DE1发展版控制系统结构进行设计，核心控制板采用Altera CycloneII系列的Nios II(EP2C20F484C7)嵌入式处理器，其发展板及规格如图15 和表2所示，此款处理器为低成本、高性能的FPGA器件，其系统性能可依据需求而自行配置，包含：  
 (1)三种Nios II处理器：快速型(NiosII/f)、标准型(NiosII/s)、经济型(NiosII/e)，采用32比特指令集结构体系；  
 (2)完整的外设硬件设置，例如：timer、bridge、counter等，配合SOPC Builder集成完整的微处理器结构；  
 (3)Avalon交换结构，并能同时对复数单元进行处理，提高系统带宽，并使用最少的FPGA资源。

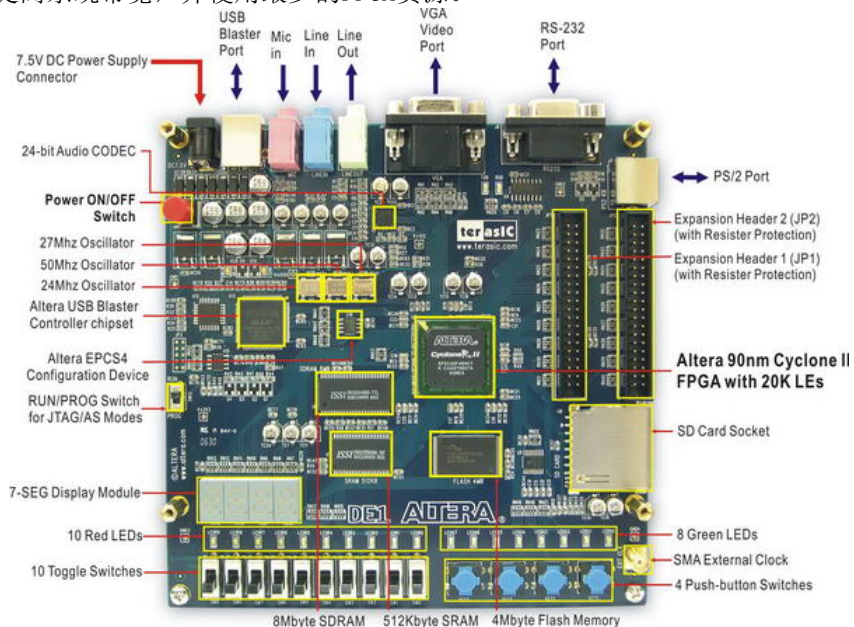


图15 DE1发展板

表2 Nios II 发展板规格表

Total Les	18752
M4K RAM blocks	52
Total RAM bits	239616
Embedded multipliers	26
PLLs	4
Maximum user I/O pins	315
FineLine BGA palage	484

设计一个Nios II系统需要分软件和硬件两方面下手，在硬件部分，利用Quartus II和SOPC软件来完成设计，可以选择使用方框图编辑器来生成上层方框图设计文档(Block Design File, .bdf)，同时也可以使用文本编辑器来设计硬件的VHDL或Verilog文件，或是直接利用内置的MagaWizard Plug-in Manager 生成较低阶的VHDL 设计文档。

在硬件设计流程中，我们使用SOPC Builder 来建立我们的Nios II系统模块，包括CPU、内存、以及其它外设电路。我们从左边选取所需要的模块加入至右边的清单中，并且设定各个模块器件的参数值，与指定其基地址(Base Address)及中断矢量(IRQ)和系统频率，如图16所示。

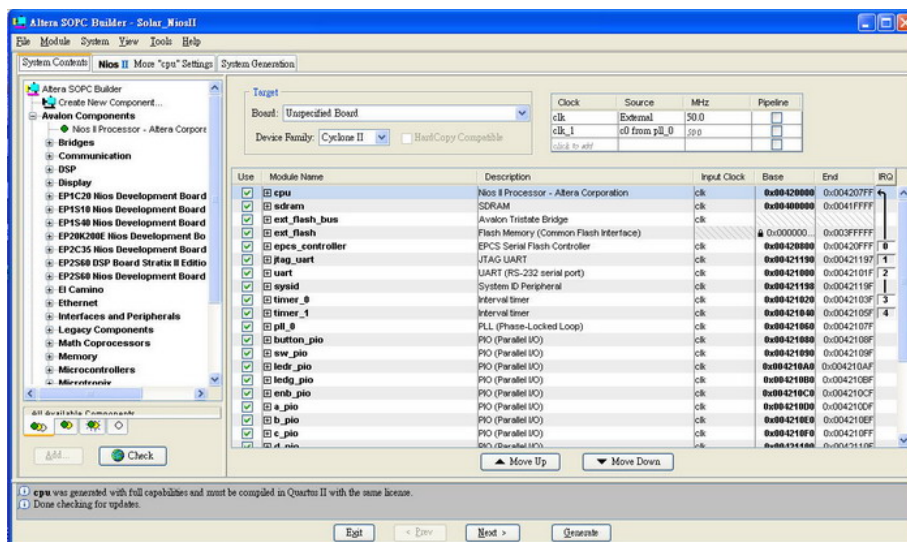


图16 SOPC Builder 的系统内容

接下来，我们必须为设计指定内存和启动(Boot)芯片、中断矢量表(Interrupt Vector Table)的位置，以及其它系统模块设定(System Module Setting)，如图17所示。

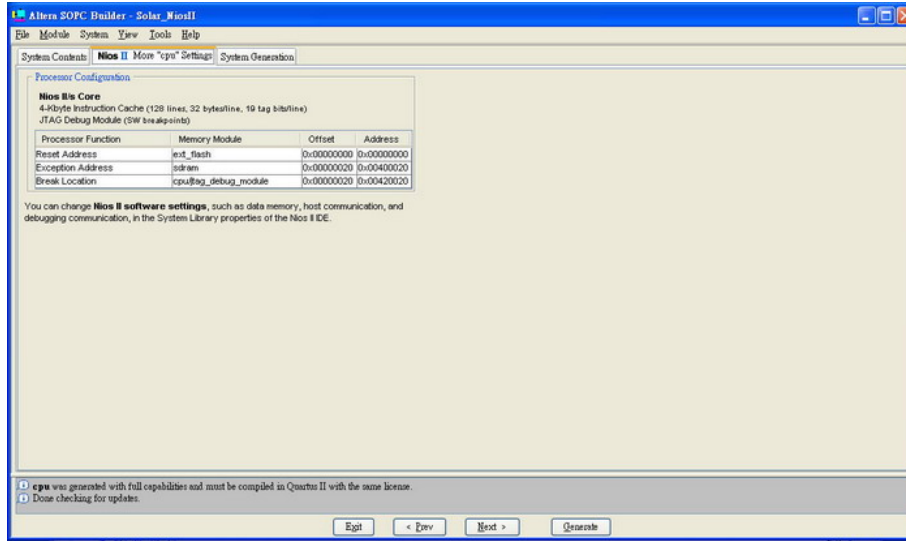


图17 NIOS II 的系统设定

完成系统设计后，把HDL 选项勾选，并选好所要配置的芯片型号，如未安装仿真软件 (ModelSim) 则不需勾选，并且点击Generate 进入生成阶段，此阶段将执行下列动作：

1. 生成HDL 原始文档。
2. 生成仿真项目和源档。
3. 生成C 和汇编语言的标头档(Header)和源档。
4. 编译系统链接库(Library)。

当SYSTEM GENERATION COMPLETED 信息出现时，这表示已经完成生成阶段，请点击Exit 选项回到Symbol 对话框，如图18 所示。

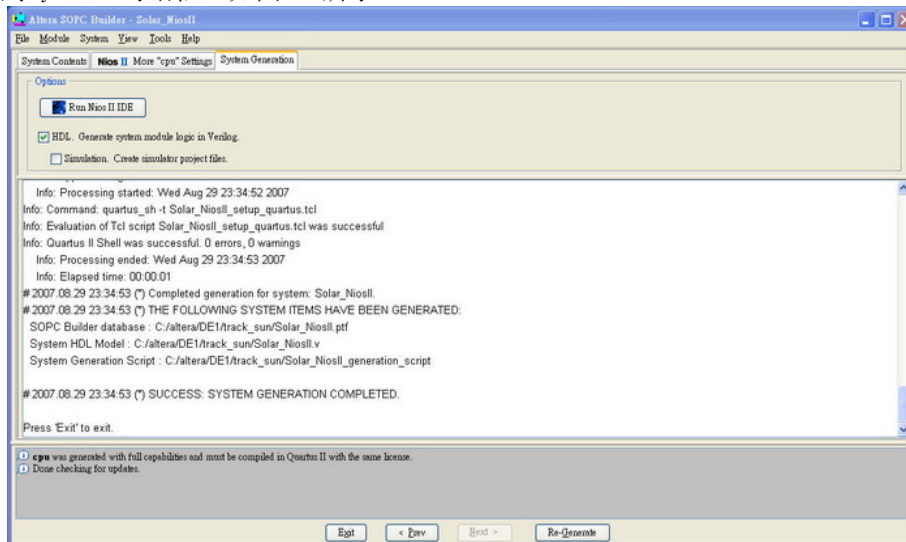


图18 SOPC 系统生成图

SOPC Builder 会生成一个系统模块的符号(Symbol), 将这个电路符号加入至BDF 文件内, 如图19 所示。接著加入输入、输出及双向脚位(Pins), 命名各脚位, 以及其它基本器件符号。

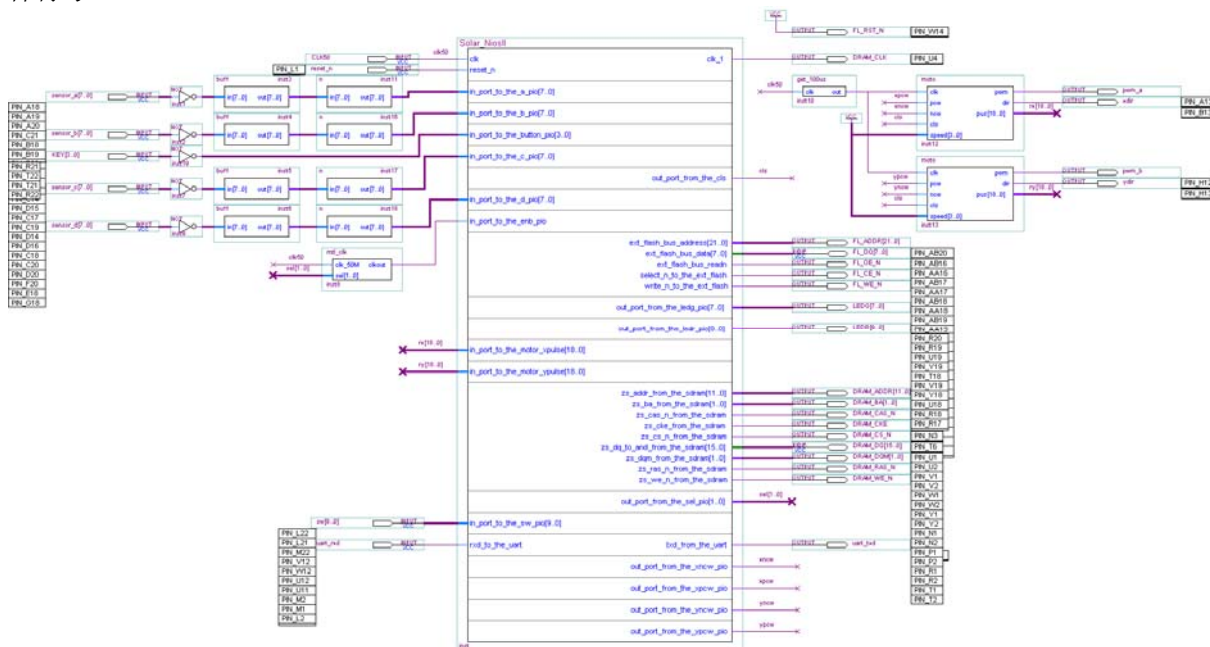


图19 Quartus II 的BDF文档

最后将所设计好电路经由Programmer Downloads至发展板上, 如图20 所示。

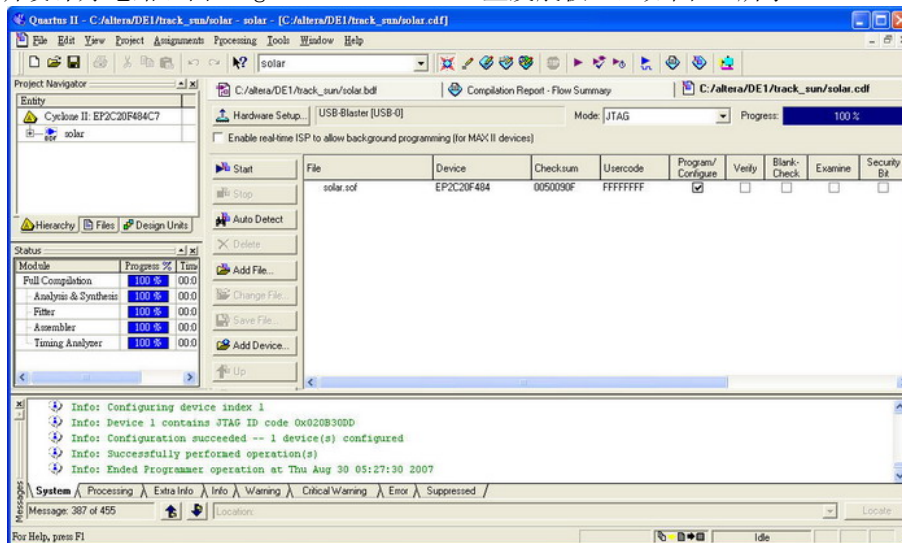


图20 Quartus II 刻录至芯片

我们利用Verilog 所编辑光敏电阻传感器输入译码电路, 此信号将送至Nios II CPU来作为控制器判断处理, 如图21 所示。

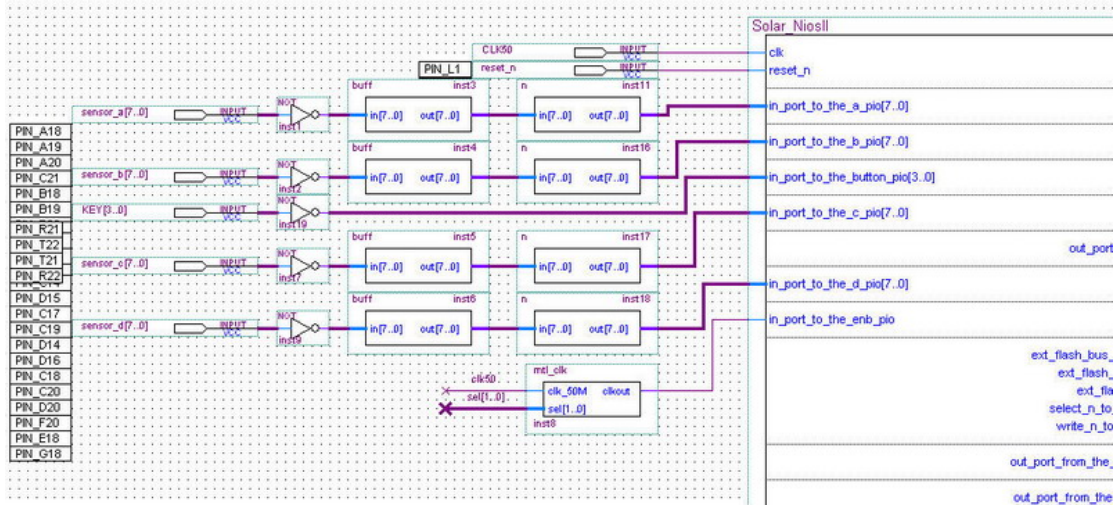


图21 光敏电阻传感器输入译码电路

而经由Nios II CPU判断完太阳能板所需要调整到达角度给予电机信号来驱动，如图22所示，可以看到有两个电机模块，分别为控制此系统X及Y轴电机区块。

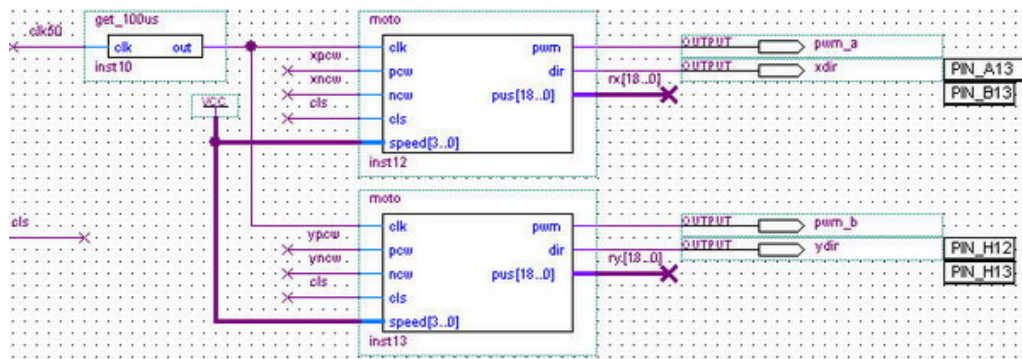


图22 步进电机模块

当生成我们所需要的系统硬件后，便可考虑将控制器用硬件或是软件来实现，因此我们考虑用系统中的Nios II -32bit CPU 来加速我们模糊控制法则，其Nios II IDE编辑器如图23所示。

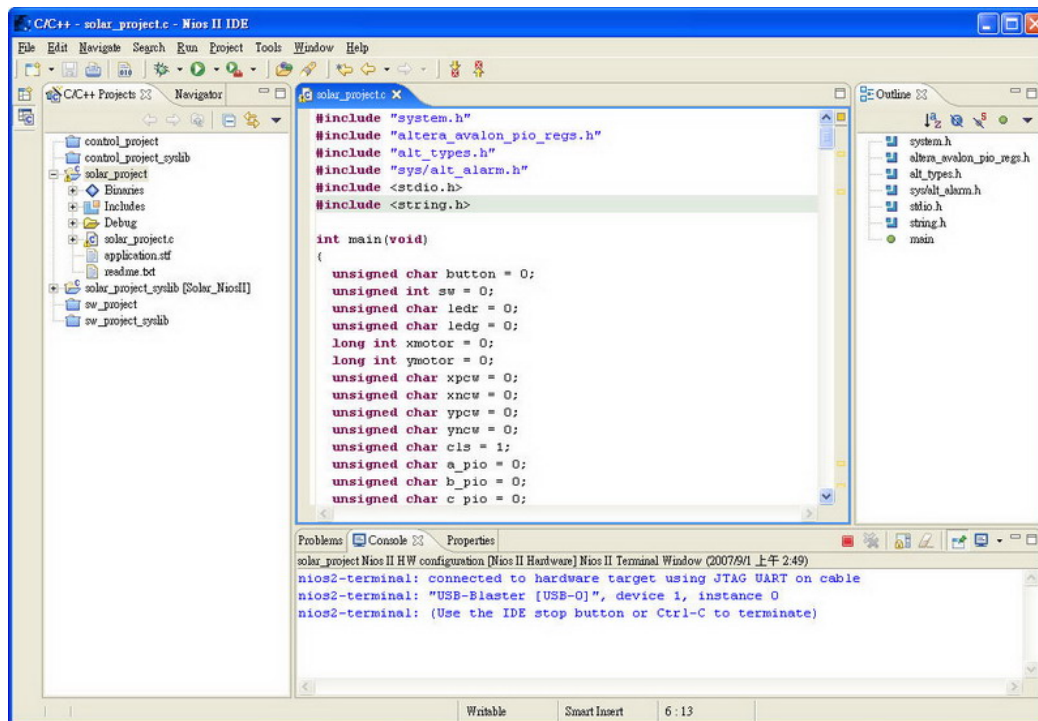


图23 Nios II IDE编辑器

## 六. 设计特点

### 1. Nios如何在本研究中提供各方面的协助

我们主要与传统单芯片8051、PIC等..最大差异在于,我们在电路上,需加入模糊控制法则,以传统芯片无法进行硬件描述语言VHDL的撰写上,因此,若使用传统芯片可能需要外在逻辑电路来辅助实现模糊控制器,然而,这样可能会造成控制器设计在体积上及成本上的负担,或者“使用具有MCU的芯片加上FPGA(不含Nios)”,需要两个芯片组合。

对于复杂逻辑电路,我们设计使用Quartus II软件在Nios II的集成开发环境(IDE)可以使用具有Nios嵌入式核心处理器之FPGA,其优点除了可以利用C语言撰写模糊算法于Nios II中,另外也可编写硬件描述语言VHDL写入FPGA中,进而控制步进电机,以达到演算法计算及I/O控制两者平行处理,以提高整体效能,并快速实现并验证我们的硬件电路。

在最后我们实际将太阳能追踪控制系统拿到户外实际测试,并以一天为测试单位,可以获得步进电机+ FPGA发展板+其余控制及感测电路是在每小时搜寻追踪一次,每次约30秒,一日约充电时间约8小时,其余时间皆不耗电(亦即不待机,不耗电),则可算出以下合理的数据,如表3:

表3 全天候太阳能辐射收集情形(多云)

测试方式	固定型太阳能集电系统	智能型太阳能集电系统
太阳能电池全日平均累计发电量 $P_s$ (J)	276480	345600
步进电机, 控制及感测电路全日累计耗电量 $P_c$ (J)	0	9216
全日平均累计净发电量 $P_{total} = P_s - P_c$ (J)	275480	336384

实验结果显示, 经过固定式仰角控制比较智能型太阳能追踪控制的总静发电量得知智能型太阳能集电系统优于固定型太阳能集电系统。

### 户外实际测量情况

本研究将太阳能平台搬至学校顶楼以固定和智能型两种测量如上表3 所示, 如图24及图25所示分别为户外固定型及智能型太阳能集电系统仿真情况。

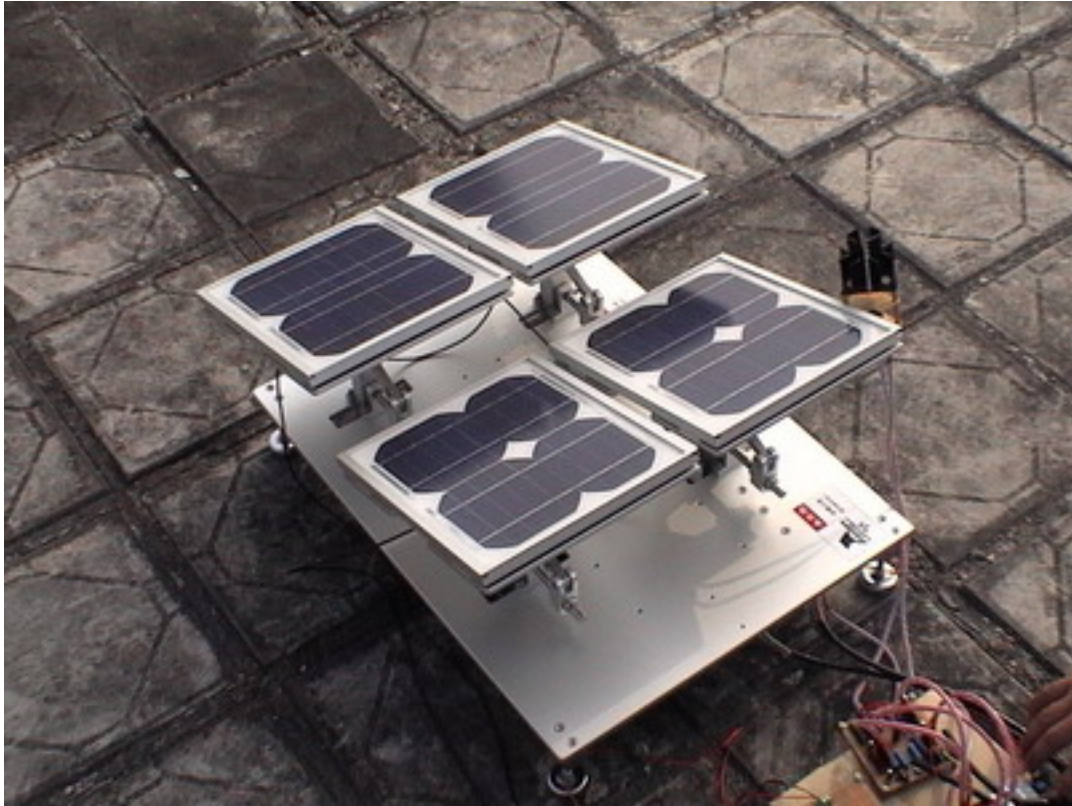


图24 户外固定型太阳能集电系统仿真情况



图25 户外智能型太阳能集电系统仿真情况

#### 室内以探照灯实际仿真太阳光照情况

本研究以探照灯为仿真太阳光源，并在我们自行架设仿真太阳行进轨道以固定型和智能型两种，并使用Visual Basic (VB)以RS232将量测电压传送至NoteBook (NB)实测电压值，可以看出相同角度上发电效率比较，如图26和图28 为室内固定型及智能型太阳能集电系统仿真情况。



图26 室内固定型太阳能集电系统仿真情况

我们使用Visual Basic(VB)以RS232将量测电压传送至NoteBook(NB)实测电压值，如图27和图29为室内固定型及智能型太阳能集电系统电压。

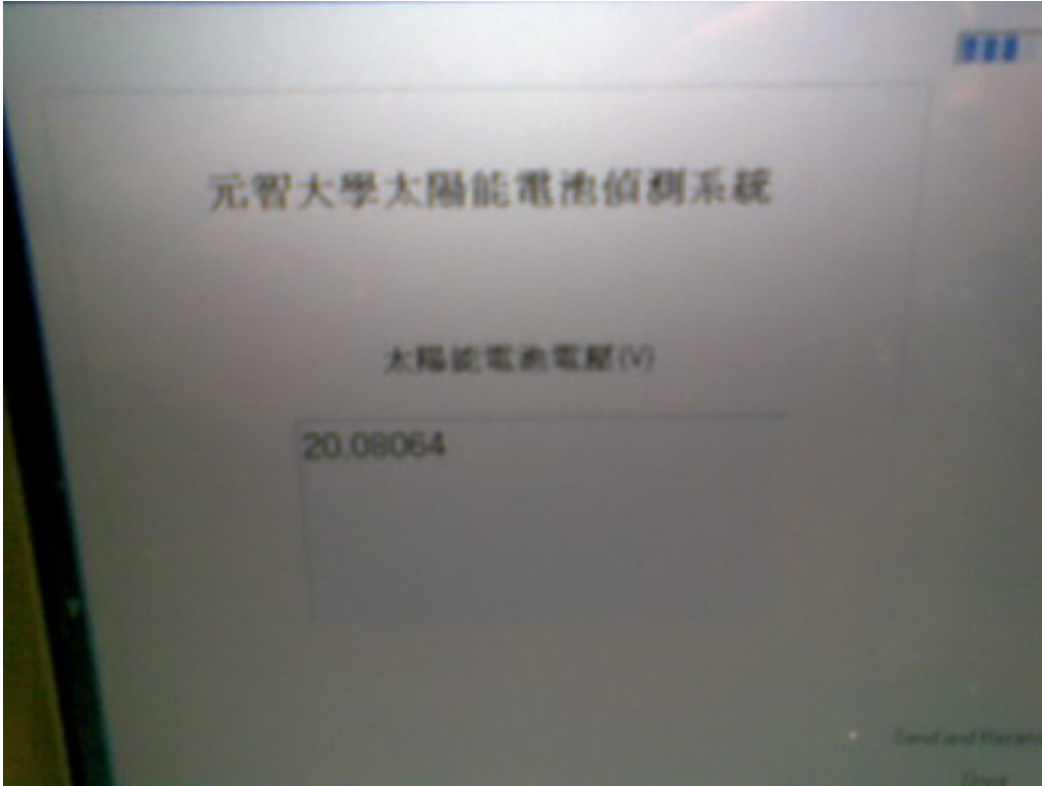


图27 室内智能型太阳能集电系统电压



图28 室内智能型太阳能集电系统仿真情况

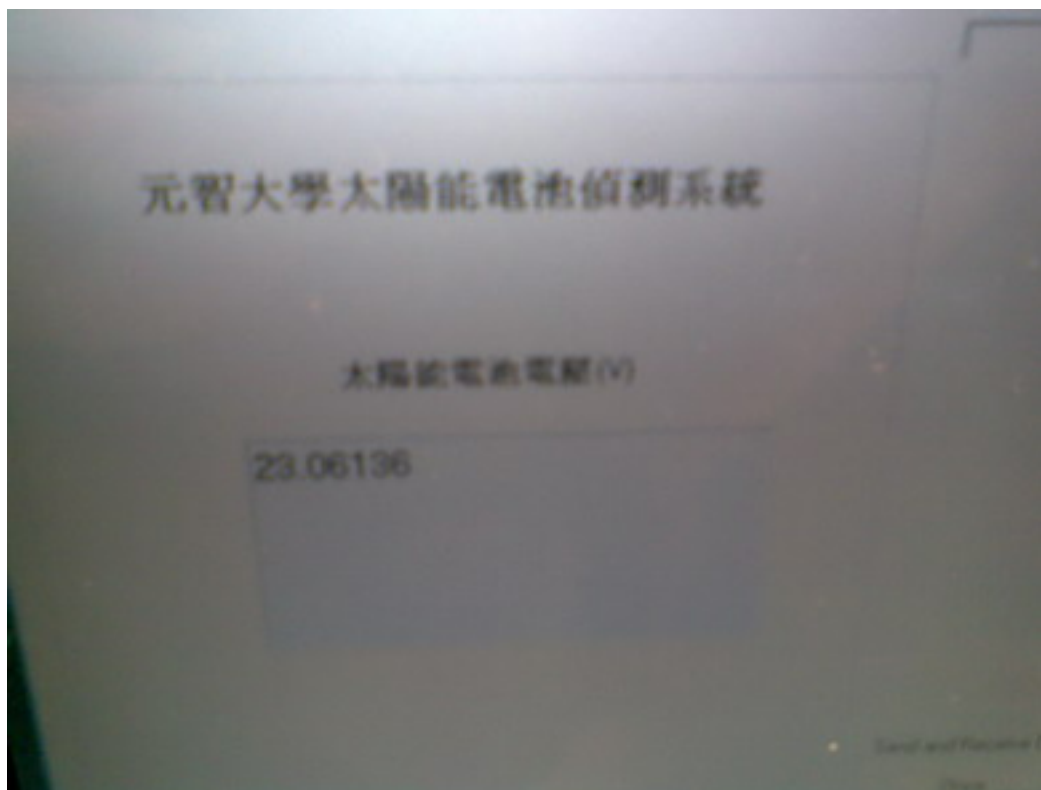


图29 室内智能型太阳能集电系统电压

最后我们由图27和图29，两张图可以看出固定型电压小于智能型太阳能集电系统，所以很清楚智能型太阳能集电系统优于固定型太阳能集电系统。

## 2. 结论

本研究的实验结果可获得以下结论：

- a. 利用光敏电阻、步进电机及 FPGA 之 Nios II 程序撰写制作出不同的太阳光的处理策略，如云层干扰处理策略和复归策略…等，则可构建出智能型仰角控制系统进行太阳光源的追踪以提高太阳能电池的发电效率。
- b. 由表 3 全天候太阳能辐射收集情形可以证实智能型仰角控制式相当值得推广的太阳能电池发电系统。
- c. 通过 Altera FPGA 之 Nios 微处理器，可以发现在研究开发及产品测试将变得较为容易且电路板可以缩小，节省许多成本，大幅优于许多单芯片。

## 七. 总结

在参加这次比赛之前，我们使用Altera公司的FPGA和CPLD相关产品，对Nios了解不陌生，不过对于Nios II微处理器的了解却是只停留在Altera公司的网站及公司宣传上。本次Nios II设计大赛给我们创造一个良好增强对使用Nios II微处理器设计的机会。通过本次比

赛，使我们对Nios II的全新设计理念和特性有了深刻的体会。

经由这次设计采用了Altera公司的SOPC系列芯片，SOPC代表一种新的系统设计技术，SOPC Builder以及Nios微处理器使我们看到软、硬件系统设计技术强大。传统电路设计大多采用PCB板上由硬件组件搭建而成，如果发现错误或者系统需要改进、升级，必须重做PCB板，若要调整和修改带来很多不便，同时也增加设计成本和开发周期。这次竞赛我们以竞赛时间内可以快速完成作品，从设计思维到实现，只需要在计算机PC上进行建模即可，ALTERA提供完整的工具，包含SOPC Builder、Nios II集成设计环境(IDE)和Quartus II发展环境，来加速软件和硬件的开发。从第一代Nios发展至第二代Nios II，大幅改善过去第一代缺点，提高了效能，使设计更为精巧，而且对于系统的稳定性也有很大的提高，且Nios II的发展、测试环境和C语言编译器的集成，大幅增进用户的便利性。