

1. 系统概述

在半导体工业中，硅晶片在从一个工序移动到下一个工序的过程中是被装在盒子里的。当盒子到达某一位置时，晶片搬运机器人就把晶片从盒子里转移到加工模块，一次一片。机器人必须跳过没有晶片的位置或者那些一个槽里不止一片晶片的位置（双槽）。为了获得这些位置信息，在机器人搬运这些晶片之前，晶片在盒子中的位置需要被定位。

这个应用实例所描述的，是一个单轴的晶片盒定位机（如图 1），这个定位机被连接到一台 RZ0 机器人（在此不做详细介绍）。机器人发出请求后，晶片定位机就驱动一个传感器对 25 槽晶片盒中的晶片堆进行扫描，并输出一个数字波形用来指示晶片边沿的位置。每个槽中晶片的状态是通过电机位置及与之对应的波形中的每个边沿确定的。当整个晶片盒被定位完毕后，晶片定位器会将哪个晶片槽里有东西及哪个槽是双槽的信息通过以太网报告给机器人控制器。

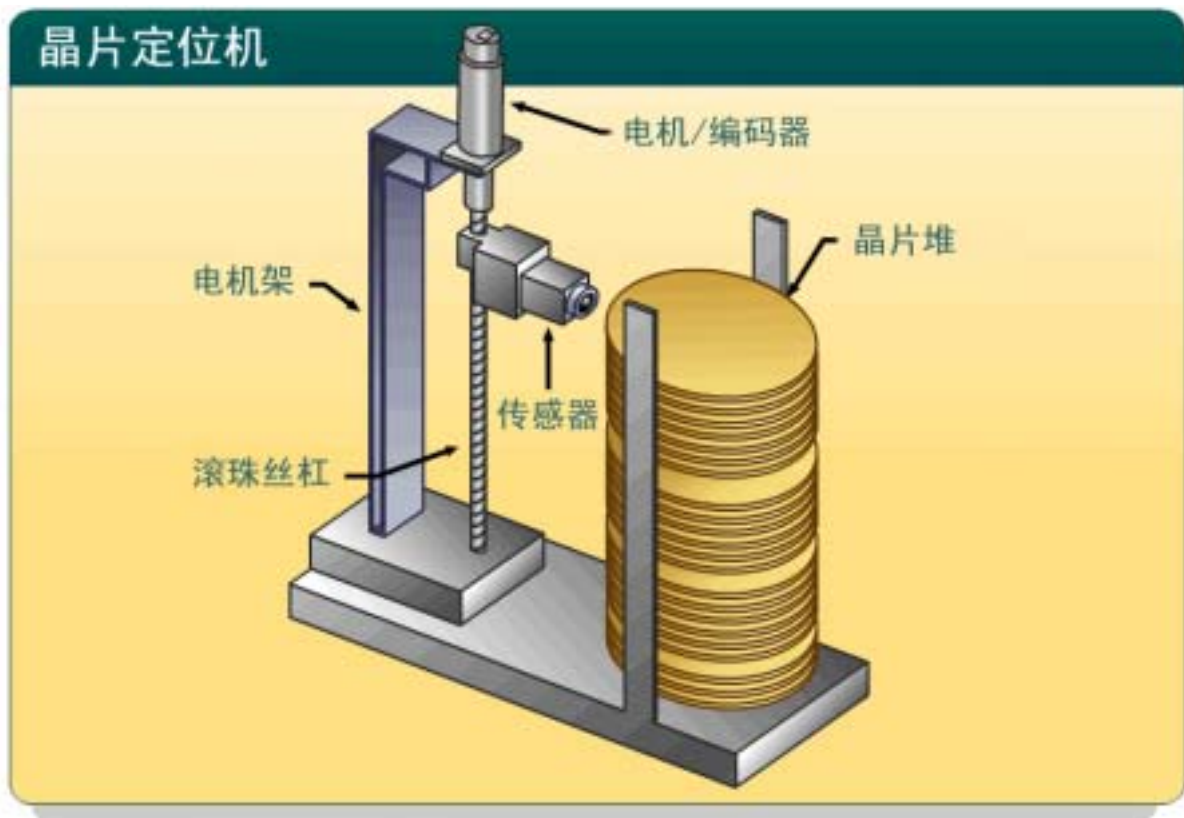


图 1. 晶片定位机（晶片盒，机器人未在图中显示）

2. 要求

- 用来捕捉晶片边沿位置的高速位置锁存器
- 从以下几种状态中判断盒子中每个槽的状态：
 - 无晶片
 - 有晶片
 - 双槽(同一槽中有一片以上的晶片)

- 通过一个以太网连接将定位结果报告给机器人控制器
- 回零开关输入

3. 部件的选择

本段描述了为实现晶片定位机控制系统，您可以选择的 GALIL 的硬件及软件产品。下面是一个主要部件的列表，后面附有简单描述。

表 1. 晶片定位器控制系统需要的材料清单

部件名称	描述	价格 (U.S.) 单买 / 批量
CDS-3310	带有以太网及 RS-232 单轴控制器及驱动系统	\$745/\$495
BLM-N23-50-1000	带有 100 线编码器及 Hall 传感器的 Nema 23 无刷伺服电机	请咨询生产商
WSDK Servo Tuning Software	伺服调试及分析软件	\$195 (one time)

控制器：CDS-3310

晶片定位机需要一个单轴用来移动传感器扫描晶片。Galil 的 CDS-3310 有刷或无刷伺服电机单轴控制器及驱动系统是个很好的选择。它将一个高性能的可编程运动控制器与一台 500W 的 PWM 驱动器集成在一起，形成一个紧凑的，高性价比的包裹。CDS-3310 控制器支持 10/100Base-T 以太网，同时也支持非易失性的程序存储器，使它成为独立式控制器应用的理想选择。



图 2. CDS-3310 控制器及驱动系统

电机：BLM-N23-50-1000

为了达到免维护运行，我们选择无刷电机。由于晶片定位机需要不到 0.3Nm 的连续转矩，所以我们推荐使用 Galil 的 NEMA 23 #BLM-N23-50-1000 无刷电机或者其他同等电机。

电机上装有一台每转 1000 脉冲的增量编码器，积分后每转产生 4000 个脉冲。由于增量编码器为输入信号提供了通向 CDS-3310 的交换通道，电机上不许要安装 Hall 传感器。

4. 功能实现

本段详细介绍了如何通过上述已选择的部件来实现控制系统。

系统的连接

晶片搬运机器人包括一个机器人（由 Galil DMC-2133 控制器控制），集线器，CDS-3310 定位控制器，和定位机（如图 3）。机器人控制器通过以太网线经由集线器与定位控制器相连接。

以太网通讯(IH,SA)

在控制器中，每一个网络连接都由一个“句柄”标号来表示，从 A 到 H。每个标号都对应于一个 IP 地址和端口号。IP(Internet Protocol)地址在网络连接里是唯一的，它用来识别网络中的一台

设备，如控制器或计算机。一旦通讯群组通过 IH 命令建立起来，计算机便可以通过 SA 命令向网络中的其他设备发送指令，并接收响应。

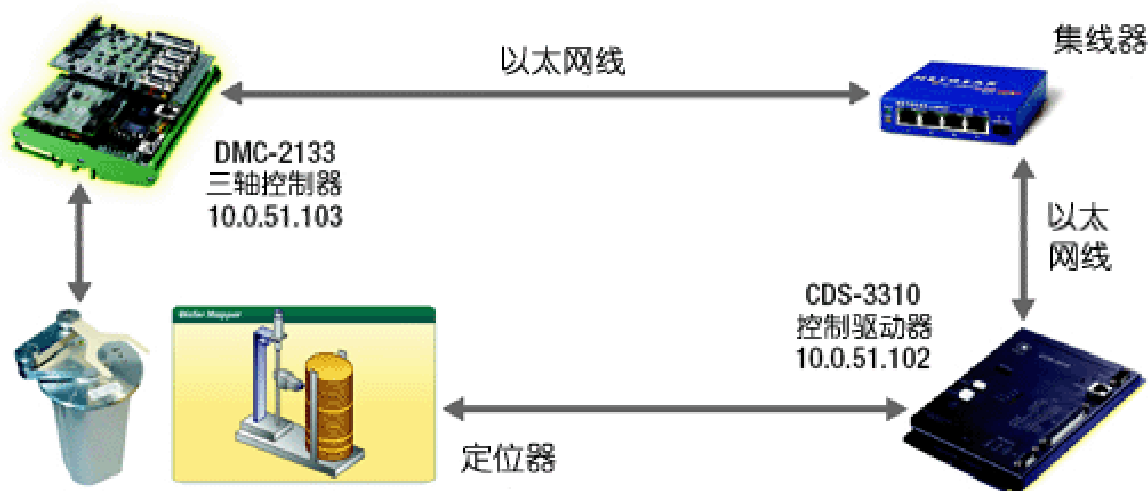


图 3. 系统连接图

IHF=10,0,51,102 ; ' 为定位器分配一个地址
SAF="XQ#map" ; ' 向定位器发送启动命令

在此例中，机器人控制器为定位机控制器分配了一个地址 10.0.51.102，并用标号 F 表示。然后机器人控制器向定位机控制器发送 XQ#map 命令，执行定位机控制器中的#map 子程序。

高速位置锁存(CN, AL, _RL)

Galil 运动控制器特有一路被称为高速地址锁存的专用数字输入信号，硬件上是一个边沿触发器。在此应用实例中，锁存特性被用来精确地判断在每个晶片边沿处轴的位置(每片晶片有两个沿)，并给出传感器输出波形，如图 4 所示。

传感器输出波形图

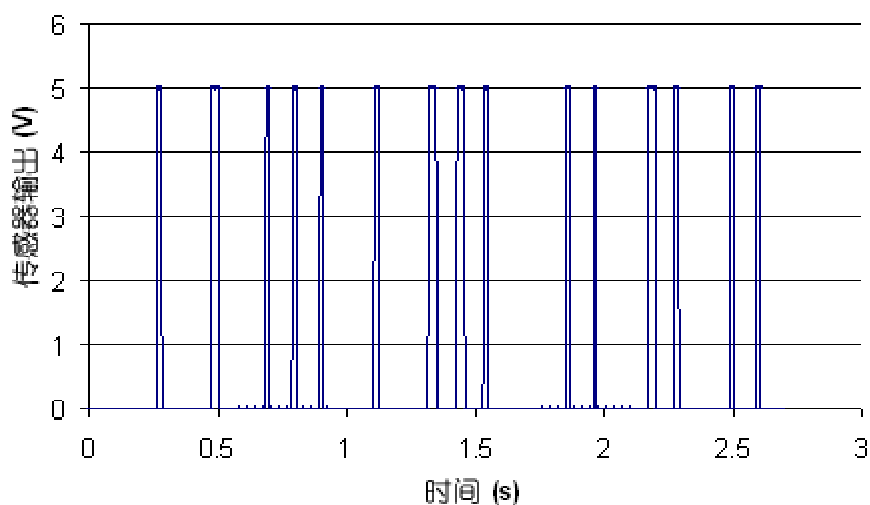


图 4. 定位机传感器输出波形图
在每个上升沿及下降沿捕捉电机的位置

当通过 `ALX` 命令启动锁存后，控制器在探测到输入状态变化的时刻会自动将编码器的位置值写入一个寄存器内。通过 `_RLX` 命令可以读取该寄存器。通过 `CN` 命令可以控制在上升沿 (`CN, , 1`) 或下降沿 (`CN, , -1`) 锁存位置。

程序描述

这里有两个程序：`robot.dmc` 和 `mapper.dmc`。`Robot.dmc` 在机器人控制器上运行，且仅包含初始化定位操作及显示定位结果的程序。`Mapper.dmc` 在 CDS-3310 上运行，用来对晶片进行定位。

两个程序都包含一个 `#AUTO` 主程序，用来在上电时自动执行一次初始化程序（类似建立通讯及回零）。

机器人控制器通过 `#strtmap` 子程序启动定位。之后 CDS-3310 控制器在晶片堆前移动传感器，同时收集传感器反馈回来的信息。当定位机控制器完成了对整个晶片盒的扫描后，它会将每个晶片槽的状态返回给机器人控制器。每个晶片的状态是以数组的形式传送给机器人的，该数组有 25 个元素（从 0 到 24）。该数组里每个元素中的数值与晶片盒中一个槽的状态相对应，对应关系如下：

- 0—槽中无晶片
- 1—槽中有一单片晶片
- 2—槽中有双槽晶片（厚度超过 200 counts）

程序输出

下面是一个定位器输出的示例，它显示了每个晶片边沿的位置：

槽： 2	起始沿： 2481	结束沿： 2617
槽： 4	起始沿： 4498	结束沿： 4807
槽： 6	起始沿： 6525	结束沿： 6660
槽： 7	起始沿： 7522	结束沿： 7657
槽： 8	起始沿： 8516	结束沿： 8654
槽： 10	起始沿： 10528	结束沿： 10664
槽： 12	起始沿： 12545	结束沿： 12854
槽： 13	起始沿： 13554	结束沿： 13843
槽： 14	起始沿： 14543	结束沿： 14687
槽： 17	起始沿： 17582	结束沿： 17713
槽： 18	起始沿： 18573	结束沿： 18712
槽： 20	起始沿： 20593	结束沿： 20901
槽： 21	起始沿： 21595	结束沿： 21735
槽： 23	起始沿： 23617	结束沿： 23748
槽： 24	起始沿： 24611	结束沿： 24748

下面是一个机器人输出的示例，它显示了从定位器上载的晶片状态数组：

槽： 0	状态： 0	槽： 10	状态： 1	槽： 20	状态： 2
槽： 1	状态： 0	槽： 11	状态： 0	槽： 21	状态： 1
槽： 2	状态： 1	槽： 12	状态： 2	槽： 22	状态： 0
槽： 3	状态： 0	槽： 13	状态： 2	槽： 23	状态： 1
槽： 4	状态： 2	槽： 14	状态： 1	槽： 24	状态： 1
槽： 5	状态： 0	槽： 15	状态： 0		
槽： 6	状态： 1	槽： 16	状态： 0		
槽： 7	状态： 1	槽： 17	状态： 1		
槽： 8	状态： 1	槽： 18	状态： 1		
槽： 9	状态： 0	槽： 19	状态： 0		

程序清单

下这里给出两个程序：robot.dmc 和 mapper.dmc. 注释中有对程序的详细说明。

```
'Robot.dmc (IP 地址 10.0.51.103)
```

```
#AUTO ;'上电自动运行
DM rcassett[25] ;'声明数组(0 无晶片, 1 单晶片, 2 双槽)
IHF=-1 ;'关闭句柄
#handle
IHF=10,0,51,102 ;'向定位器打开句柄
WT1000
JP#handle, _IHF2 <> -2 ;'等待句柄开启
EN

#strtmap ;'定位动作起始
SAF="XQ#map" ;'发送命令, 起始定位动作
mapping=1 ;'定位进行中标志
#wait;JP#wait, mapping=1 ;'循环直至定位结束

'显示定位结果:
VF10,0 ;'设置变量显示格式
slot = 0 ;'清除循环变量

#list
MG "Slot:", slot, " Status:", rcassett[slot] ;'显示数据
slot = slot + 1 ;'循环变量自增
JP#list, slot < 25 ;'循环
EN
```

```
'Mapper.dmc (IP 地址 10.0.51.102)
```

```
#AUTO ;'上电自动运行
DM cassette[25] ;'定义数组

IHH=-1 ;'关闭句柄
#handle
IHH=10,0,51,103 ;'对 DMC-2133 机器人控制器打开句柄
WT1000
JP#handle, _IHH2 <> -2 ;'等待句柄开启

VF10,0 ;'设置变量显示格式
maxthick = 200 ;'单片晶片允许厚度最大值
pitch = 1000 ;'每个槽(晶片边沿)对应的编码器脉冲数
```

```

'回零到反向限位并寻找标志脉冲:
JGX = -10000      ;'Jog 方式向反向限位逼近
BGX              ;'运动开始
AMX              ;'等待运动完成

JGX = 500        ;'设置一个寻找标志脉冲的缓慢速度
FIX              ;'寻找标志脉冲
BGX              ;'运动开始
AMX              ;'等待运动完成

PR3665           ;'从第一片晶片向下偏移半个槽的距离(500 脉冲)
BGX              ;'运动开始
AMX              ;'等待运动完成
DP0              ;'定义零点
EN

#map
'将数组元素初始化为零 (无晶片状态)
count = 0        ;'循环变量赋初值
#loop
    cassette[count] = 0 ;'将每个数组元素设为零
    count = count + 1
JP#loop, count < 25

SPX=10000        ;'设置速度
ACX=1000000     ;'设置加速度
DCX=1000000     ;'设置减速度
PA25000         ;'设置目标位置
BGX              ;'x 轴开始运动

'这个循环捕捉每个晶片起始沿及结束沿的位置
#record
    CN,,1        ;'将锁存输入设置为高电平有效
    ALX          ;'启动锁存
    #wait1;JP#wait1,(_BGX & _ALX=1) ;'等待起始沿触发锁存
    JP#movefin,_BGX=0 ;'动作结束时退出
    lead=_RLX    ;'晶片起始沿的位置

    CN,,-1       ;'将锁存输入设置为低电平有效
    ALX          ;'启动锁存
    #wait2;JP#wait2,(_BGX & _ALX=1) ;'等待结束沿出发锁存
    JP#movefin,_BGX=0 ;'动作结束时退出
    trail=_RLX   ;'晶片结束沿的位置

    slot=@INT[lead/pitch] ;'当前的槽数

```

```

MG "Slot:", slot, " Lead:", lead, " Trail:", trail

IF ((trail - lead) <= maxthick)
    cassette[slot] = 1 ;'有晶片
ELSE
    cassette[slot] = 2 ;'双槽晶片
ENDIF
JP#record          ;'循环

#movefin
PA0                ;'移回零点
BGX                ;'运动开始
AMX                ;'等待运动完成

;'将数组传输给机器人控制器:
SAH="rcount=",0;WT4 ;'初始化机器人控制器上的远端记数值
count=0            ;'初始化定位控制器上的本地记数值

#loop2
    SAH="rcassett[rcount]=", cassette[count]; WT4 ;'发送数组元素
    SAH="rcount=rcount+1"; WT4 ;'累加机器人控制器上的远端记数值
    count = count + 1 ;'累加定位控制器上的本地记数值
JP#loop2, count < 25

SAH="mapping=",0; WT4 ;'设置机器人上的变量通知定位完成
EN

```