Utilización de un Microcontrolador como interfase para la conexión a Internet de una Celda Robotizada

Guillermo Lio[1], *Gustavo Placer*[1,2]

[1] *Universidad Nacional General Sarmiento, Buenos Aires, Argentina*
[2] *Universidad Interamericana, Buenos Aires, Argentina*
glio@ungs.edu.ar

Resumen

Este trabajo presenta la primera parte de un desarrollo que se está realizando en la Universidad Nacional de general Sarmiento, Laboratorio de Ingeniería para la integración de una celda de manufactura robotizada a Cloud Computing (Computación en la nube) [1] utilizando un microcontrolador como interface.

El conocimiento de aplicaciones con microcontroladores nos permite integrar en una celda robotizada un control que puede remplazar a un PLC (Controlador lógico programable) y así hacer el control de la celda por un lado y además hacer la gestión de todos los datos de producción para luego ser integrados a Cloud Computing.

Este trabajo muestra una de las diversas formas de conectividad de señales previsibles para el desarrollo de celdas de manufactura en el ámbito local. Como objetivos puntuales queremos mostrar la utilización del robot IRB 120 [2] disponible en la Universidad Gral. Sarmiento en operaciones de manufactura utilizando tecnología con microcontroladores.

Esta integración se utilizará para simular una pequeña celda de manufactura con la aplicación de "Picking" (Tomar y guardar cargas ordenadamente) [3] ubicada en el laboratorio de robótica de la Universidad UNGS.

Se simulará la selección y almacenamiento de una pieza como usualmente se realiza en condiciones de trabajo real. El control de las señales para que el robot realice las diferentes trayectorias serán establecidas por un microcontrolador MC9S08SH8CPJ de Motorola [4], donde se programarán todas las rutinas de control y gestión de la célula.

1. Introducción

Las industrias de manufactura, entre otras, podrían adquirir cada vez más células robotizadas flexibles [5]. Las mismas consisten principalmente en 1 o más robots y un autómata programable para el control y supervisión de las tareas de los robots, así como también para gestionar todos los requerimientos del operador, carga de recetas, etc.

El costo de estas células puede disminuirse en parte utilizando un microcontrolador junto con electrónica para sus correspondientes adaptaciones de señal para llevarlo a un ambiente industrial donde todo convive en 24 VCC., pero las grandes ventajas que podemos obtener a la hora de elegir esta propuesta es que nosotros mismos nos fabricamos un pseudo PLC, liberándonos de la exclusividad que tiene la industria de los PLC industriales donde todo es propietario y poco flexible, ya que uno no puede salir de los estándares de las marcas, donde los módulos, cabeceras, protocolos de comunicación, software de programación, firmware, etc. no son intercambiables y ni permiten ser modificados por los usuarios. Cada modificación en estos sistemas implica altos costes para las pequeñas y medianas empresas.

Fig. 1 – Comunicación Robot-Microcontrolador-CloudComputing.

Las prestaciones de un microcontrolador son similares a las de un autómata programable. La única desventaja es que su nivel de programación requiere un poco más de trabajo ya que se programa en lenguaje assembler, C o C++, en comparativa con un PLC que programan en ladder o function blocks que son lenguajes de alto nivel. Respecto a la robustez de los PLC al ruido industrial podemos obtener excelentes resultados utilizando fuentes de alimentación filtradas y entradas salidas optoacopladas.

El objetivo de este trabajo es demostrar de manera práctica el reemplazo de un autómata por un microcontrolador (Fig. 1) para controlar un brazo robótico de 6 ejes para una aplicación de picking (tomar y acomodar cajas) [3].

2. Conexionado

El microcontrolador utilizado es el MC9S08SH8CPJ de Motorola [4], el brazo robótico que se controló es un ABB IRB120 [2].

Las salidas del Microcontrolador se configuraron en el PTA y las entradas en el PTB. La tensión de trabajo es de 5VCC.

El microcontrolador lo conectamos a la placa DSQC 652 del Robot [6] que es una placa industrial de 16 Entradas/Salidas. La misma trabaja con 24VCC.

Para poder intercambiar señales previamente hubo que hacer una adaptación de las mismas.

El conexionado eléctrico desde el Microcontrolador al Robot se muestra en la Fig. 2.

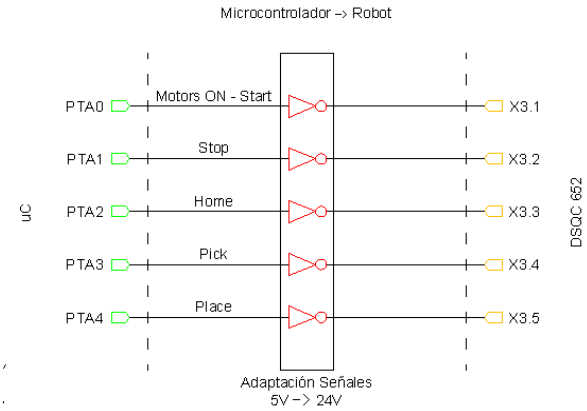

Fig. 2 – Conexión desde el Microcontrolador al Robot

Por otro lado, el conexionado eléctrico entre el Robot y el Microcontrolador se muestra en la Fig. 3.

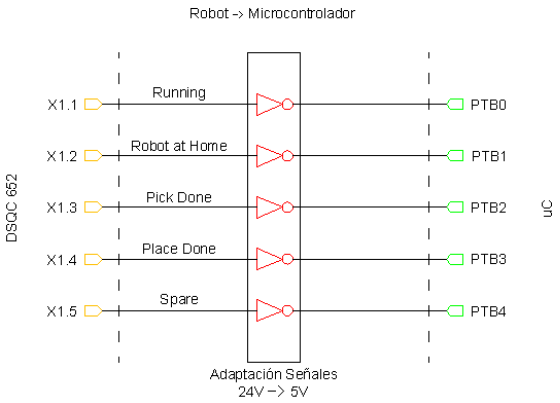

Fig. 3 – Conexión desde el Robot al Microcontrolador

3. Funcionamiento

El funcionamiento típico en una celda de manufactura se ha descripto para controlar la celda robotizada.

Todos los datos de producción serán guardados en bases de datos internas del microcontrolador para luego poder disponer de los datos y así ser enviados a la nube.

Podemos ver el funcionamiento a partir de un diagrama de flujos Fig. 4.

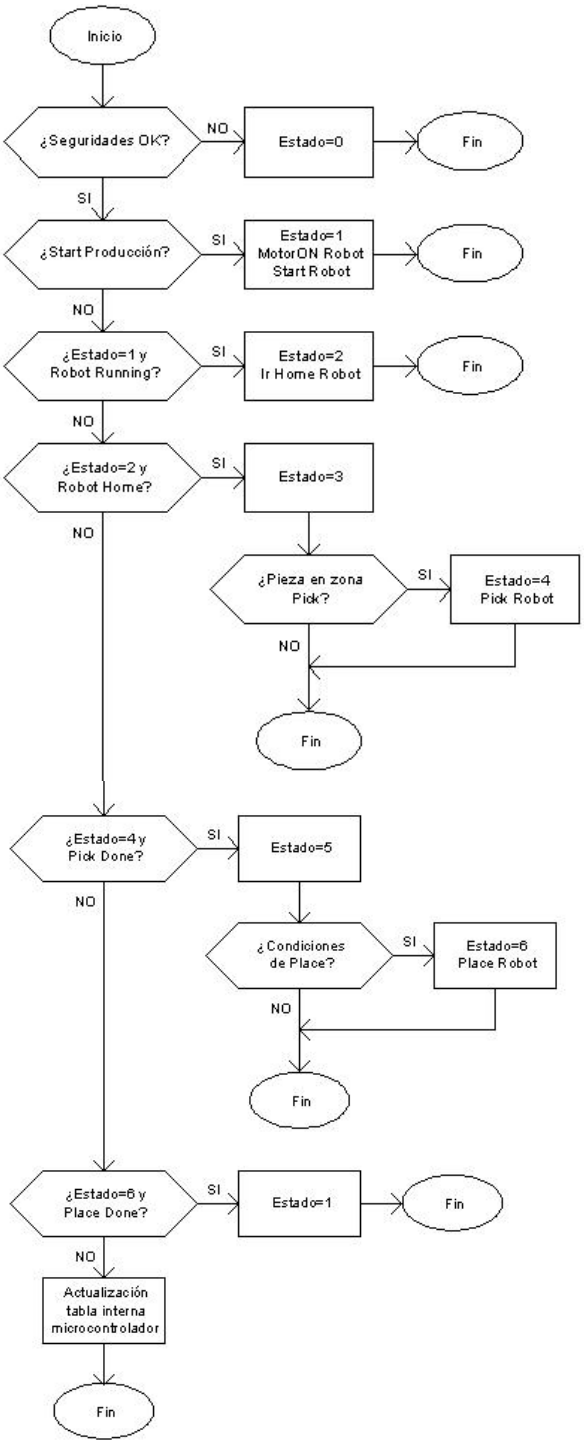

Fig. 4 – Diagrama flujo Microcontrolador

4. Lazo principal programa robot

El robot se programa en un pseudo C llamado RAPID [7]. El lazo principal de programa del Robot (main) queda a la espera de las órdenes del microcontrolador. Ni bien llega una orden, el robot ejecuta su trayectoria de movimiento ya sea la RobHome, RobPick o RobPlace. Estas rutinas sólo tienen instrucciones de movimiento. Luego de ejecutada la rutina de movimiento da aviso al microcontrolador que la posición fue alcanzada mediante una señal de salida.

```
PROC main()

        IF di_Home = 1 THEN
                RobHome; !ejecuta rutina de
                movimiento a posición Home
                Setdo do_Robot_at_Home, 1;
        ENDIF

        IF di_Pick = 1 THEN
                RobPick; !ejecuta rutina de
                movimiento a posición Pick
                Setdo do_Pick_Done, 1;
                WaitTime 2;
        ENDIF

        IF di_Place = 1 THEN
                RobPlace; !ejecuta rutina de
                movimiento a posición Place
                Setdo do_Place_Done, 1;
                WaitTime 2;
        ENDIF

        WaitTime 0.2;

ENDPROC
```

5. Lazos Microcontrolador

```
void main(void) {
  EnableInterrupts;
  asm_main();  /* call the assembly function */

  PTADD=PTADD | 0xFF;//Puerto A como salida
  PTAD=0;//Inicializa el Puerto A en cero
  PTBDD=PTBDD | 0x00;//Puerto B como entrada
  PTBD=0;//Inicializa el Puerto B en cero

  for(;;) {

        PickExecute();
        PlaceExecute();
        AccesExecute();
        ValidateExecute();
```

```
  __RESET_WATCHDOG() ;/* feeds the dog */
 } /* loop forever */
}

//Condiciones Para Pick de Pieza
void PickExecute(void) {
	// Si el Robot esta en Running y hay pieza en pocisión de Pick
	if (DiRunning==0x01 & DiSensorPick==0x01)
	{
		DoPick=0x01; //Orden de Pick al Robot
		//Espera señal de Pick realizado correctamente
		while (DiPickDone==0x00 & (TimeOut < TimeOutMax))
		{
			TimeOut = TimeOut + 0x01;
		}
		if (TimeOut >= TimeOutMax)
		{
			//Salida de Error de tiempo de espera agotado
			//Se pasa el Robot a Stop
			DoErrorTimeOut = 0x01;
			DoStop = 0x01;
		}
		//Se limpian comandos
		TimeOut = 0x00;
		DoPick=0x00;
	}
}

//Condiciones Para Place de Pieza
void PlaceExecute(void) {
	//Si el Robot esta en Running, el Pick se realizó correctamente
	//y la pocisión de Place esta vacía
	if (DiRunning==0x01 & DiPickDone==0x01 & DiSensorPlace==0x00)
	{
		DoPlace=0x01; //Orden de Place al Robot
		//Espera señal de Place realizado correctamente
		while (DiPlaceDone==0x00 & (TimeOut < TimeOutMax))
		{
			TimeOut = TimeOut + 0x01;
		}
		if (TimeOut >= TimeOutMax)
		{
			//Salida de Error de tiempo de espera agotado
			//Se pasa el Robot a Stop

			DoErrorTimeOut = 0x01;
			DoStop = 0x01;
		}
		//Se limpian comandos
		TimeOut = 0x00;
		DoPlace=0x00;
	}
}

//Condiciones Para Acceso Celda
void AccesExecute(void) {
	//Si el Robot esta en Running y se hizo peticiòn de acceso
	//a la celda
	if (DiRunning==0x01 & DiPulsadorAcceso==0x01)
	{
		DoHome=0x01; //Orden de Robot a pocisión de Home
		//Espera señal del Robot en Pocisión de Home
		while (DiRobotAtHome==0x00 & (TimeOut < TimeOutMax))
		{
			TimeOut = TimeOut + 0x01;
		}
		if (TimeOut >= TimeOutMax)
		{
			//Salida de Error de tiempo de espera agotado
			//Se pasa el Robot a Stop

			DoErrorTimeOut = 0x01;
			DoStop = 0x01;
		}
		//Se limpian comandos
		TimeOut = 0x00;
		DoHome=0x00;
	}
}

//Condiciones Para Arranque Celda
void ValidateExecute(void) {
	//Validación de celda o Rearme
	if (DiPulsadorValidacion==0x01)
	{
		//Limpia registros
		DoStop = 0x00;
		DoErrorTimeOut = 0x00;
		//Arranque de Motores y Play de Programa
		DoMotorsOnStart=0x01;
```

```
                //Espera señal del Robot en Estado
Running
                while (DiRunning==0x00 & (TimeOut
                < TimeOutMax))
                {
                        TimeOut = TimeOut + 0x01;
                }
                if (TimeOut >= TimeOutMax)
                {
                        //Salida de Error de tiempo de
                        espera agotado
                        //Se pasa el Robot a Stop

                        DoErrorTimeOut = 0x01;
                        DoStop = 0x01;
                }
                //Se limpian comandos
                TimeOut = 0x00;
                DoMotorsOnStart=0x00;

                //Si no hubo Errores en el Arranque
                if (DoErrorTimeOut==0x00)
                {
                        DoHome=0x01;  //Orden de
                        Robot a pocisión Home
                        While
(DiRobotAtHome==0x00 & (TimeOut< TimeOutMax))
                        {
                                TimeOut = TimeOut
                                + 0x01;
                        }
                        if (TimeOut >= TimeOutMax)
                        {
                                //Salida de Error de
                                tiempo de espera
                                agotado
                                //Se pasa el Robot a
                                Stop

                                DoErrorTimeOut =
                                0x01;
                                DoStop = 0x01;
                        }
                        //Se limpian comandos
                        TimeOut = 0x00;
                        DoHome=0x00;
                }
        }
}
```

6. Resultados

Se implementó una aplicación de Picking en el Laboratorio de la Universidad, simulando una pequeña celda de manufactura donde se probó el funcionamiento de la lógica descripta por el microcontrolador y por el robot para tomar cajas y ordenarlas. Se enviaron las órdenes en paralelo y el robot las interpretó describiendo un funcionamiento típico de esclavo en una celda.

Por otro lado se crearon tablas internas en el microcontrolador para así obtener una base de datos donde se almacenan todos los datos de producción, como cantidad de piezas producidas, tiempos de ciclo, recetas, distintos tipos de paradas, etc.

7. Conclusiones

La información que ahora se encuentra almacenada en una base de datos en el interior del microcontrolador, en una segunda etapa la información de las tablas será volcada a la nube, permitiendo que la información sea accesible por varios usuarios usando un navegador o utilizando una interfaz apropiada para soportar las características específicas de la aplicación.

La computación en la nube no requiere ningún hardware específico y se puede ejecutar sobre una estructura con acceso a Internet, una computadora personal, notebook o teléfonos inteligentes como Iphone o Android.

En definitiva, mientras que en la actualidad existe una tendencia a recurrir a sistemas pesados o robustos para resolver una aplicación industrial que en muchos casos podría ser resuelta en forma mucho más económica, en este trabajo se muestra que con tecnologías de microcontroladores se pueden desarrollar soluciones igual de efectivas y de menor coste para la industria pequeña o mediana así también como para fines educativos [8].

8. Referencias

[1] Cearley, D. W. (2010), “Cloud computing”

[2] Robot Industrial ABB IRB 120 http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/3bd625bab3c7cae1c1257a0800495fac/$file/ROB0149EN_D_LR.pdf

[3] Escudero Serrano, , Logística de almacenamiento M. J. (2014)

[4] MC9S08SH8CPJ de Motorola Data Sheet, 28/40-Pin 8-Bit CMOS FLASH Microcontrollers, disponible en www.alldatasheet.es/datasheet- pdf/pdf/346352/ FREESCALE/ MC9S08SH8CPJR.html

[5] http://celdas-de-manufactura.es.tl/CELDAS--DE-MANUFACTURA.htm

[6] DSQC 652
http://www.abb.com/productdetails/ABB.PARTS.SEROP3HAC025917-001

[7]https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RAPID_3HAC16581-1_revJ_en.pdf

[8] http://perso.wanadoo.es/pictob/microcr.htm