

AR1795 Robotics Vision System

Robotics Vision System

Project Number - AR1795

Abstract

The Robotics Vision System (Figure 1) is designed with on the Philips LPC2138 microcontroller and the HAMAMATSU CMOS linear image sensor S9226. The microcontroller is programmed with software that implements a very efficient algorithm for detection of a line position in the camera's object plane and converts the line position in analog voltage. The vision system can be used in many robotics applications where position detection is necessary. That includes line following toys and industrial robots. The system is also perfect for distance measuring by the triangulation method. The goal of this robot is develop a small, cost efficient vision system to be used on Robofest and other robot competitions.



Figure 1. Robotics Vision System

Block Diagram

The internal structure of the system is very simple. It consists of LPC-H2138 header board for the Philips LPC2138 microcontroller from Olimex Ltd (www.olimex.com), HAMAMATSU CMOS linear image sensor S9226 (<http://www.hamamatsu.com/>) and the objective lens.

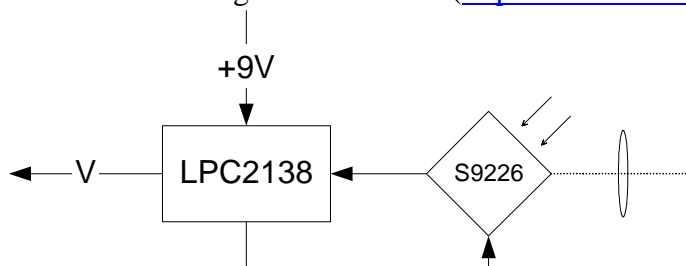


Figure 2. Block diagram

Schematic

The schematic of the Robotics Vision System is shown in Figure 3. The powerful set of peripherals of the LPC2138 and microcontroller-oriented interface of the image sensor allows glueless integration of the system. The control inputs of the sensor connected directly to the capture inputs and the match output of the microcontroller timers.

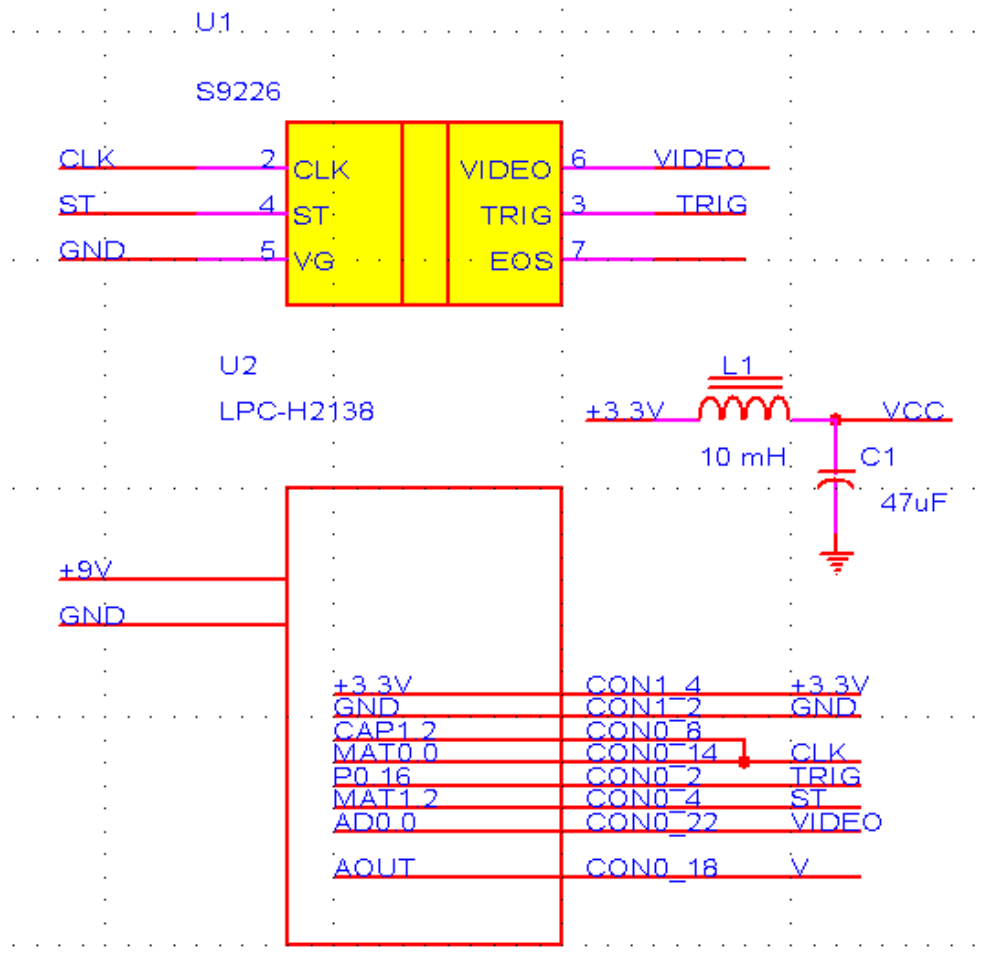


Figure 3. The schematic of the Robotics Vision System

Software Implementation

The Robotics Vision System software was implemented in C++ using a GNU compiler and CrossStudio from Rowley Associates Ltd. The source code and workspace is included in the contest entry submission. A sample of the source code is presented on the listing below.

AR1795 Robotics Vision System

```
/*
*****
File:          main.cpp
Description:   main function of the application
Class:        none
Author:       AR1795 (10/14/05)
Copyright (C) 2005 AR1795
*****
*/

#include "Camera.h"
#include "Filter.h"

extern "C" int main(void);
int main (void)
{
    CCamera Camera;
    CFilter Filter;

    __ARMLIB_enableIRQ();
    while(true)
    {
        /* start image capturing */
        Camera.Start();
        /* wait till the image is ready */
        while (false == Camera.IsReady());
        /* calculate image position */
        unsigned short wLinePosition = Filter.Calculate((const unsigned char*)CCamera::s_vbyPixels,
        CCamera::IMAGE_SIZE);
        /* output line position as voltage */
        DACR = (wLinePosition & 0x3FF) << 6;
    }
    return 0;
}
```