Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com

Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

PropCAM-DB (#28320)

The PropCAM-DB is a daughterboard that gives black-and-white (grayscale) vision to Parallax Propeller-based systems. It is directly plug-compatible with both the Parallax Propeller Backpack (#28327) and the Parallax Spinneret (#32203). With the Parallax DB-Expander (#28325) adapter, it will also accommodate other Propeller host systems.

NOTE: This product comprises a *limited edition* of 1500 units total. The package label indicates how far into the total edition each unit represents. This information is provided to help customers plan for the PropCAM's inevitable end-of-life.

Features

- 128 x 96-pixel digital grayscale (8-bit) image sensor, using as few as 6144 bytes of Propeller RAM for an image (4-bit pixels)
- 3.6mm focal-length lens
- Fast real-time imaging and individual snapshots, with shutter speeds down to 200 microseconds
- Simultaneous exposure of all pixels: no "rolling shutter" artifacts
- Direct connection to Propeller chip: no intermediate controller required
- Uses Kodak KAC9360 sensor, whose full datasheet is readily available without a non-disclosure agreement, permitting user modification of open-source driver firmware

Specifications

- Power Requirements: 3.0 to 3.6 VDC @ 40 mA (average)
- Communication Interface: Digital 4-bit parallel and 8-bit serial
- Operating temperature: -49 to +185 °F (-45 to +85 °C)
- Dimensions: 1.35 x 1.35 x 1.50 in. (35 x 35 x 38 mm)

Application Ideas

- Vision-based inspection systems
- Motion detection
- Networkable webcam (using Spinneret)

Additional Items Required

- Propeller P8X32A host system: Propeller Backpack (#28327), Spinneret Web Server (#32203) or, with the DB-Expander (#28325), any other Propeller-based host system
- Source of 3.3 V power (usually provided by the host system)
- Video monitor (Backpack) or web browser (Spinneret) for viewing the acquired images

Quick-Start Guide

Propeller Backpack

Load the program **PropCAM_autoexposure_demo.spin** into the Propeller Backpack's EEPROM, power down, then plug in the PropCAM-DB and an NTSC monitor. Make sure the "VID" jumper is installed on the Backpack and that the "OVL" jumper is not. Power up the Backpack and monitor. You will see the real-time video output from the PropCAM, along with some colored text, on the monitor screen, as shown in the photo below:

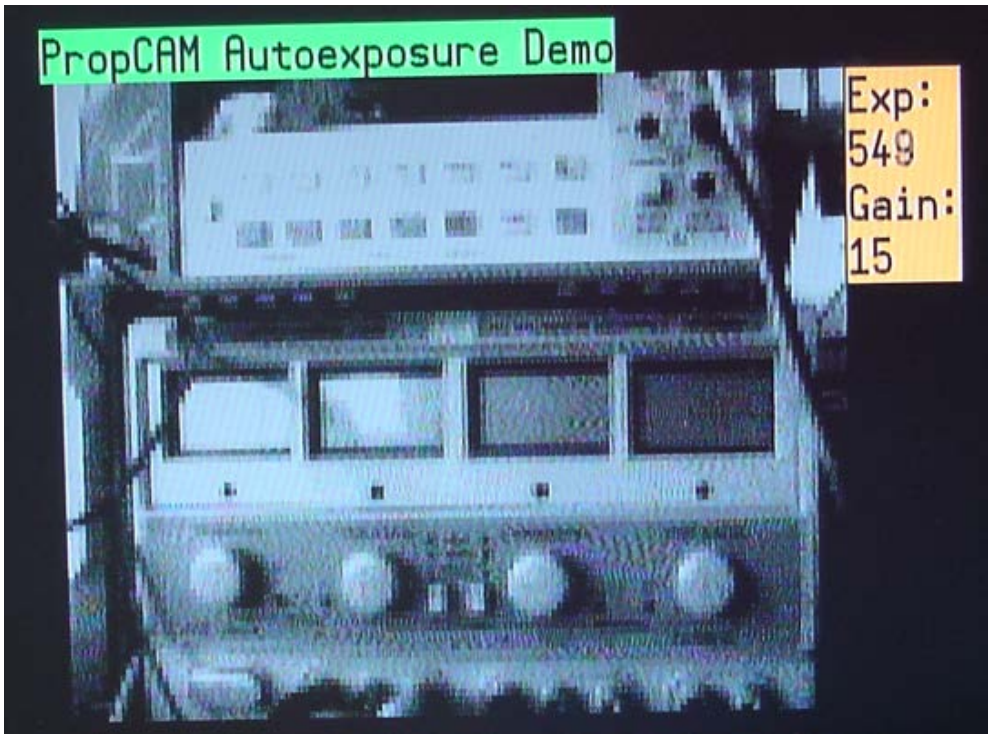

Figure 1: Auto-exposure Demo TV Screen Photo

This demo uses the 4-bit parallel PropCAM driver, **PropCAM_parallel4.spin** (16 gray levels), which packs two pixels per byte into a grayscale video buffer. It may also use, without modification, the 4-bit serial driver, **PropCAM_serial4.spin**. The 128 x 96 pixels thereby occupy only 6144 bytes of the Propeller's 32 KB RAM. The driver object accomplishes auto-exposure by continually adjusting the sensor's gain and exposure time to maintain a constant average brightness. The included **tv_gray_overlay** object works in conjunction with the **TV** object to overlay the grayscale image atop the **TV** object's display. It accomplishes the gray-level display by using a DUTY-mode output from one of the Propeller's counters and resistively coupling it into the video output. A 330 pF capacitor filters the DUTY-mode noise to provide a smoother display without killing the color or blurring details. These extra components are part of the Backpack module but may be added to other Propeller boards if needed.

Spinneret Web Server (#32203)

Load the program **PropCAM_webcam_demo.spin** into the Spinneret's EEPROM, power down, then plug in the PropCAM-DB and the Ethernet cable. Power up again, and direct your Ethernet-connected web browser to:

http://192.168.0.50:3456

(…or to whichever alternative IP address you've modified the program to respond to). You should see a webpage with a still image from the camera, which gets refreshed every two seconds, and data showing the current gain, exposure time, and average pixel values:

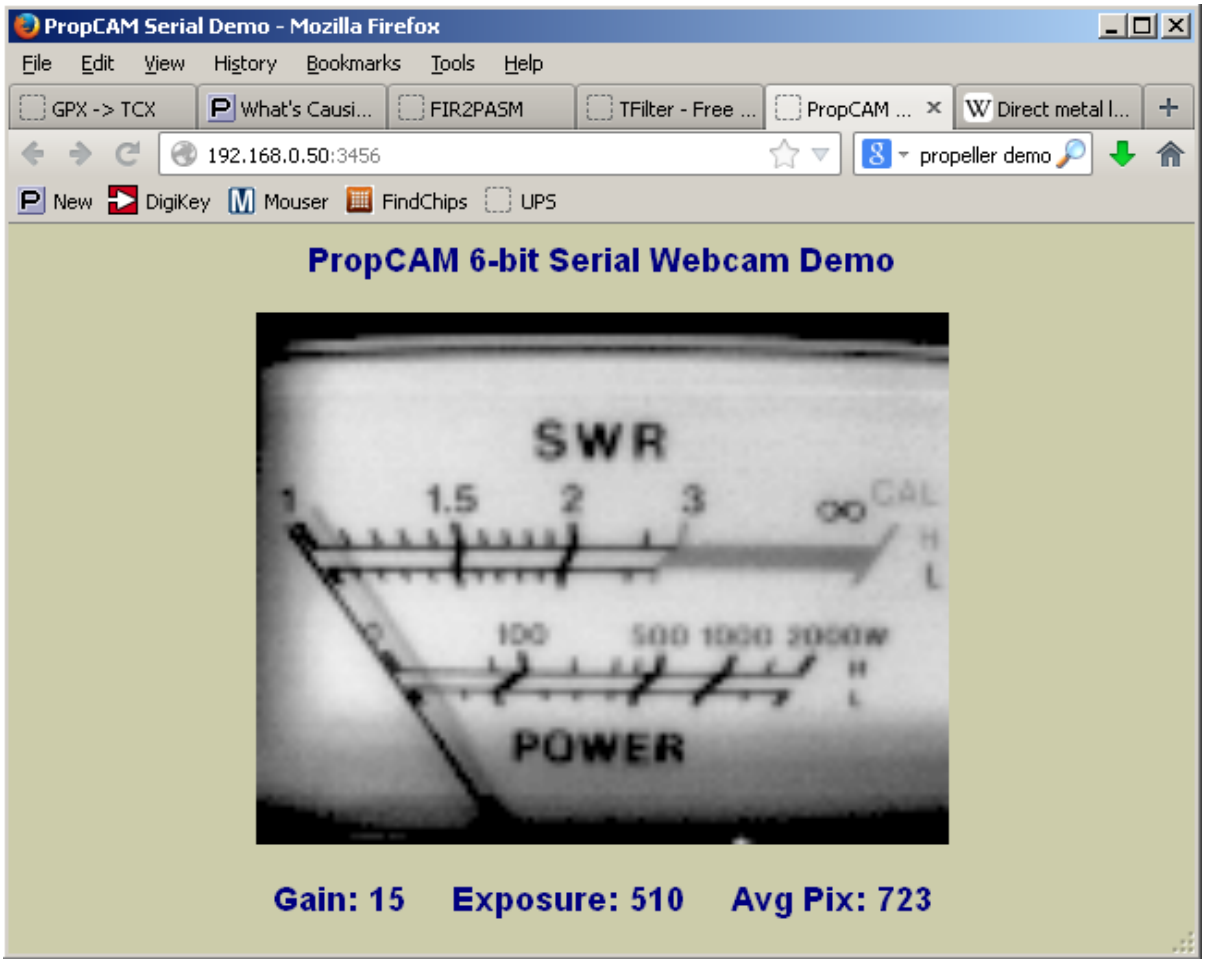

Figure 2: Webcam Demo Screen Shot

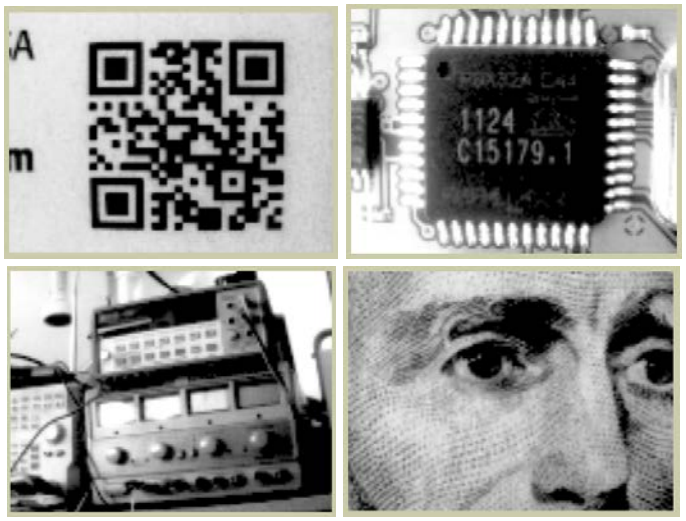

Figure 3: More Sample Images from the Webcam Demo

The image in the webpage gets displayed at three times the actual 125 x 96 image size, so each camera pixel comprises a 3 x 3 block of on-screen pixels. Depending upon your web browser, the outer pixels of adjoining blocks may be blended to provide smoother gray-level transitions, leading to a somewhat blurry appearance.

This demo uses the 6-bit serial driver, **PropCAM_serial6.spin**, for 64 gray levels and packs five pixels per 32-bit long. (That's the reason each line is 125 pixels long, instead of the full 128.) The resulting grayscale buffer occupies 9600 bytes of the Propeller's 32 KB RAM. The demo may also use, without modification, the 4-bit serial driver, **PropCAM_serial4.spin**. For transmission to the user's web browser, the demo program converts the grayscale-buffered image to a Windows bitmap (.bmp) before sending it on. Since bitmap images start at the bottom row of pixels, rather than at the top, you will see the refreshed images scanning from bottom to top.

NOTE: When used with the Spinneret, the PropCAM's SDA and SCL are shared with the EEPROM's SDA and SCL. This will usually not cause a conflict, since the sensor chip has a different address from that of the EEPROM. However, locks will be necessary if the two devices are accessed from different cogs, in order to avoid conflict. Also, the Spinneret's serial pins are shared with the PropCAM's CLK and VSYNC pins, so it will not be possible to communicate with the Spinneret via a PropPlug while the PropCAM is in operation.

Other Propeller Systems

Shown below are the minimum configurations for both parallel and serial operation of the PropCAM-DB with the Parallax Propeller P8X32A. All signal levels *to* the PropCAM are 0 to 1.0 V (logic 0) and 2.0 to 3.3 V (logic 1). Signal levels *from* the PropCAM are 0 V (logic 0) and 3.3 V (logic 1). **SDA** should be pulled up to +3.3V via a 4.7K resistor. If the host board does not provide one, the PropCAM's onboard pull-up can be used by jumpering J2 (to the right of the lens holder). The 10 MHz clock requirement can be met by an output from one of the Propeller chip's counters, configured for NCO. (A PLL counter will not be satisfactory, since it won't necessarily be synchronized to the system clock.)

In the parallel configuration, Propeller I/O pins **P0** to **P3** *must* be assigned to the camera's data outputs. The Propeller Backpack demo program uses the parallel driver object, which can be used with other parallel configurations.

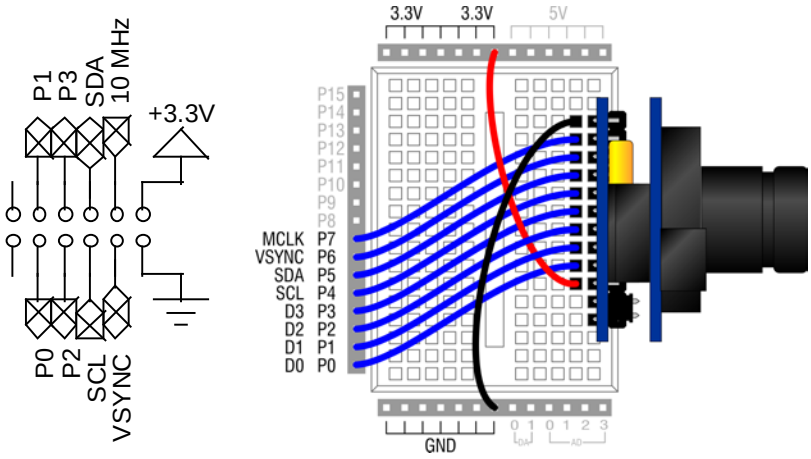

Figure 4: Parallel Interface and Example Connection

The above figure shows the connections looking down on the PropCAM connector from the top side, as well as an example connection to a Propeller Board of Education (#32900) or a Propeller Activity Board (#32910), using a DB-Expander (daughterboard-to-SIP adapter, #28325). This adapter also provides the required pull-up on SDA, so it is not necessary to provide one separately or to jumper the PropCAM's J2 thru-holes.

In the serial configuration, any Propeller I/O pins can be used for the various signals. The Spinneret demo program uses the 6-bit serial driver object, which can be used with other serial configurations.

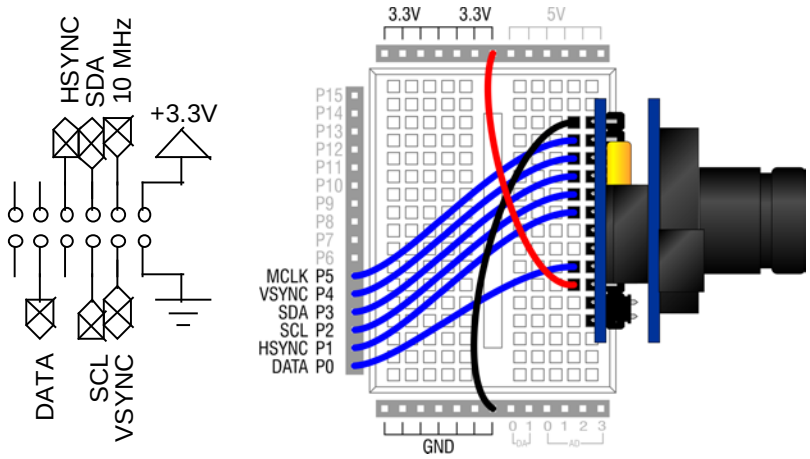

Figure 5: Serial Interface and Example Connection

The above figure shows the connections looking down on the PropCAM connector from the top side, as well as an example connection to a Propeller Board of Education (#32900) or a Propeller Activity Board (#32910), using a DB-Expander (daughterboard-to-SIP adapter, #28325). This adapter also provides the required pull-up on SDA, so it is not necessary to provide one separately or to jumper the PropCAM's J2 through-holes.

Theory of Operation

Overview

The PropCAM-DB is a module that mates Kodak's KAC9630 black-and-white 128 x9 6 CMOS image sensor to Parallax's P8X32A Propeller microcontroller. In operation, the KAC9630 requires a continuous 10 MHz clock, which the Propeller provides via one of its counters in NCO mode. The sensor also requires an I2C interface, through which several modes of operation are enabled by writing the chip's internal registers.

For data acquisition, the KAC9630 offers both parallel and serial pixel data access. In parallel mode, each pixel is output on pins D7 to D0, for eight bits of resolution, allowing 256 gray levels. The PropCAM-DB has access to the four MSBs, D7 to D4 (relabeled D3 to D0), for 16 gray levels. In this mode, two pixels can be packed per byte, so one scan will occupy 6144 bytes of the Propeller's 32 KB RAM.

In serial mode, all eight bits of each pixel are output in sequence on pin D0. This pin connects, via a 680 ohm resistor, to the KAC9630's D4 pin (renamed D0 on the PropCAM module), which tri-states in serial mode. The sensor chip's D1 pin serves as a byte synchronization marker in serial mode. This pin connects, via another 680-ohm resistor, to the chip's D7 pin (renamed D3 on the PropCAM), which also tri-states in serial mode. The PropCAM's serial drivers use it for horizontal synchronization (via the extra delay between pulses) in lieu of the KAC9630's own HSYNC output. The PropCAM's six-bit serial driver uses the six most-significant of the eight data bits for 64 gray levels and packs five pixels in each 32-bit long. This requires 9600 bytes to store a full frame.

Capture Modes

The KAC9630 sensor offers both snapshot and video (continuous) image acquisitions. The PropCAM drivers use the sensor's snapshot mode only, since it offers the most control over image synchronization. The drivers can then simulate continuous mode via a separate cog that repeatedly triggers the sensor on a user-selectable basis. The drivers support **SNAPSHOT** (capture on demand only), **CONTINUOUS** (repeat as fast as possible), **TIMED** (repeat at a fixed interval), and **SYNCED** (repeat, triggered by TV overlay vertical sync) modes. In all of these modes, no actual capture to the image buffer will take place unless requested. At other times, the driver will go through the motions of capture, accumulating average pixel data, but excluding the actual writes to the image buffer. Here's a flow chart that illustrates the process:

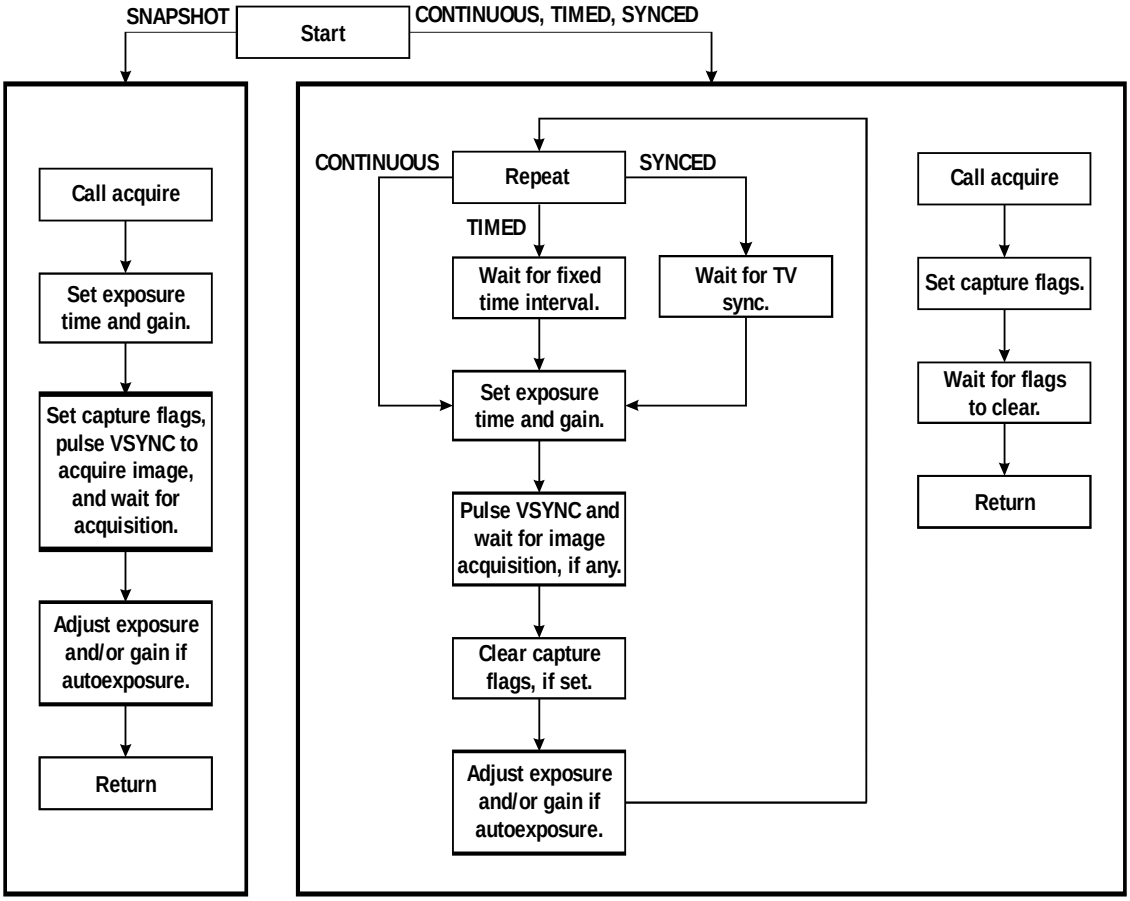

Figure 6: Image Capture Flowchart

Image Acquisition

In the driver objects, the even scan lines and odd scan lines require separate cogs to capture pixels from the sensor and write them to the Propeller's hub RAM. This is due to the 10 MHz data rate coming from the sensor (one nibble or bit every other instruction at 20 MIPs). In most cases, the even lines get written to the even lines in the Propeller's grayscale RAM buffer; the odd lines, to the odd lines in the buffer. However, the drivers also make it possible to write even scan lines to odd buffer lines — and *vice-versa* — and to capture only the even or odd scan lines. By saving the even scan lines to the even buffer lines in one capture and the same even scan lines to the odd buffer lines in the next capture, one can discern any changes that occurred between scans by comparing adjacent buffer lines pixel-by-pixel. This makes it easy not only to detect motion, but also to identify where in the field of view it occurred.

To capture pixels to the image buffer, you must call the **acquire** method with flags that indicate which scan lines to acquire and how to acquire them. The options are:

- **GRAB_EVEN**: Acquire the even pixel lines.
- **GRAB_ODD**: Acquire the odd pixel lines.
- **GRAB**: **GRAB_EVEN | GRAB_ODD** i.e. grab both even and odd lines.
- **EVEN_TO_ODD**: Write the even pixel lines to the odd buffer lines.
- **ODD_TO_EVEN**: Write the odd pixel lines to the even buffer lines.
- **LOOP_EVEN**: Grab the even pixel lines every time through the loop.
- **LOOP_ODD**: Grab the odd pixel lines every time through the loop.
- **LOOP**: **LOOP_EVEN | LOOP_ODD** i.e. grab all pixel lines every time through the loop.

The loop flags apply only to the non-**SNAPSHOT** modes of operation. All flags may be Boolean ORed to obtain different combinations of operations. For example, **GRAB_EVEN | EVEN_TO_ODD** is what you would use to grab the even pixel lines to the odd image buffer lines.

Exposure Control

Two things control the brightness of the acquired image: gain and exposure time. The higher the gain setting and/or the longer the exposure time, the brighter the resulting image will be. Both of these factors are controlled by settings in the **params** array, whose address is sent to the **start** method. The relevant values are:

- **EXP_TYPE**: one of **FIXED**, **AUTOEXP**, **AUTOGAIN**, or **AUTOBOTH**
- **EXP_TIME**: the length of an exposure in tens of microseconds (20 – 2047: 200 µs – 20.47 ms)
- **GAIN**: the log gain of the sensor's internal video amplifier (0 – 31: 1x – 28.1x)
- **TARGET**: the desired average pixel value (x100) when an **AUTOxxxx EXP_TYPE** is chosen (always normalized to four-bit pixel depth: 0–1500).

You can establish these values at startup by pre-filling the **params** array. You can also change them on the fly by calling the **set_exp** method. (You should not write these values directly to **params** after starting in any of the **LOOP** modes, unless **EXP_TYPE** is **FIXED**.)

When the **EXP_TYPE** parameter is set to **FIXED**, the **EXP_TIME** and **GAIN** settings remain at whatever you set them to. When you select one of the **AUTOxxx** types, after each exposure, the program will attempt to adjust the selected parameter (exposure time and/or gain) to keep the average pixel value (x100) as close to the **TARGET** value as possible. In the **AUTOxxx** modes, the current exposure time and gain can be read back from the **params** array, as is done in the demo programs.

Optics

The PropCAM comes equipped with a 3.6 mm focal-length lens. This produces an angular field of view that is approximately 40° wide by 30° high. In terms of actual dimensions, the field of view will be about 0.72-times-subject-distance horizontally and 0.54-times-subject-distance vertically. See example below:

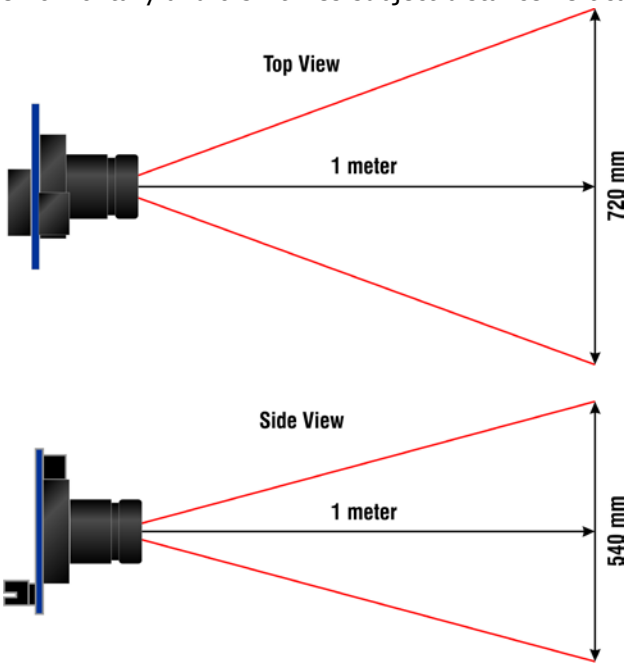

Figure 7: Horizontal and Vertical Fields of View at One Meter Distance

The lens focus is adjustable. To focus on a far object, screw the lens into its holder. To focus on a near object, unscrew the lens.

Specifications

Symbol	Quantity	Minimum	Typical	Maximum	Units
Vdd	+3.3 V supply	3.0	3.3	3.6	V
Idd	Vdd supply current	0.6	40	43	mA
VIH	Input logic “high” voltage	2.0	3.3	3.3	V
VIL	Input logic “low” voltage	0.0	0.0	1.0	V
MCLK	Main system clock frequency	8	10	10	MHz

Absolute Maximum Ratings

Symbol	Quantity	Value	Units
Vdd	Supply Voltage	4.2 max.	V
VIO	Voltage on any I/O pin	-0.3 to +4.2	V

NOTE: Absolute Maximum values are those above which actual damage to the sensor will likely occur. They do *not* indicate a safe operating range. See the Specifications chart above for safe operating conditions.

Pin Definitions

Pin	Type	Function
+5V	P	Not used unless required by mezzanine connector (J1)
+Vin	P	Not connected
D0	O	In parallel mode, KAC9630’s D4 pin; in serial mode, serial data
D1	O	In parallel mode, KAC9630’s D5 pin
D2	O	In parallel mode, KAC9630’s D6 pin
D3	O	In parallel mode, KAC9630’s D7 pin; in serial mode, HSYNC
SC	I	I2C clock (SCL)
SD	I/O	I2C data (SDA)
VS	I/O	Vertical sync (VSYNC)
CK	I	10 MHz master clock (MCLK)
+3.3V	P	+3.3V supply (Vdd)
Gnd	G	Ground
SDPU	I	Pull-up for SDA when jumpered to Vdd

Pin Type: P = Power, G = Ground, I = Input, O = Output

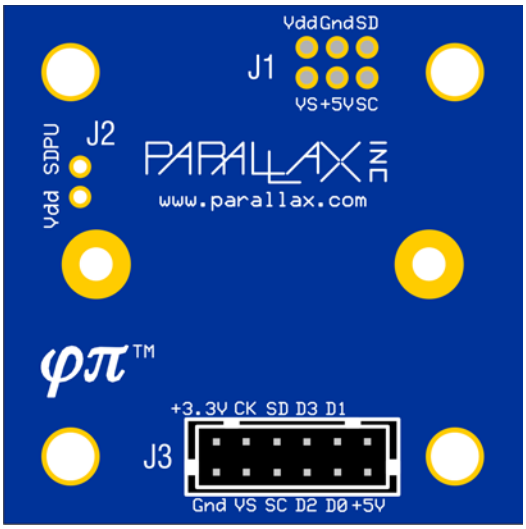

Figure 6: Rear View of the PropCAM PCB with Labeled Connections

Resources and Downloads

Check for the latest version of this document, free software, the KAC9630 image sensor datasheet, and example programs from the PropCAM-DB product page. Go to http://www.parallax.com and search for **28320**.

Schematic

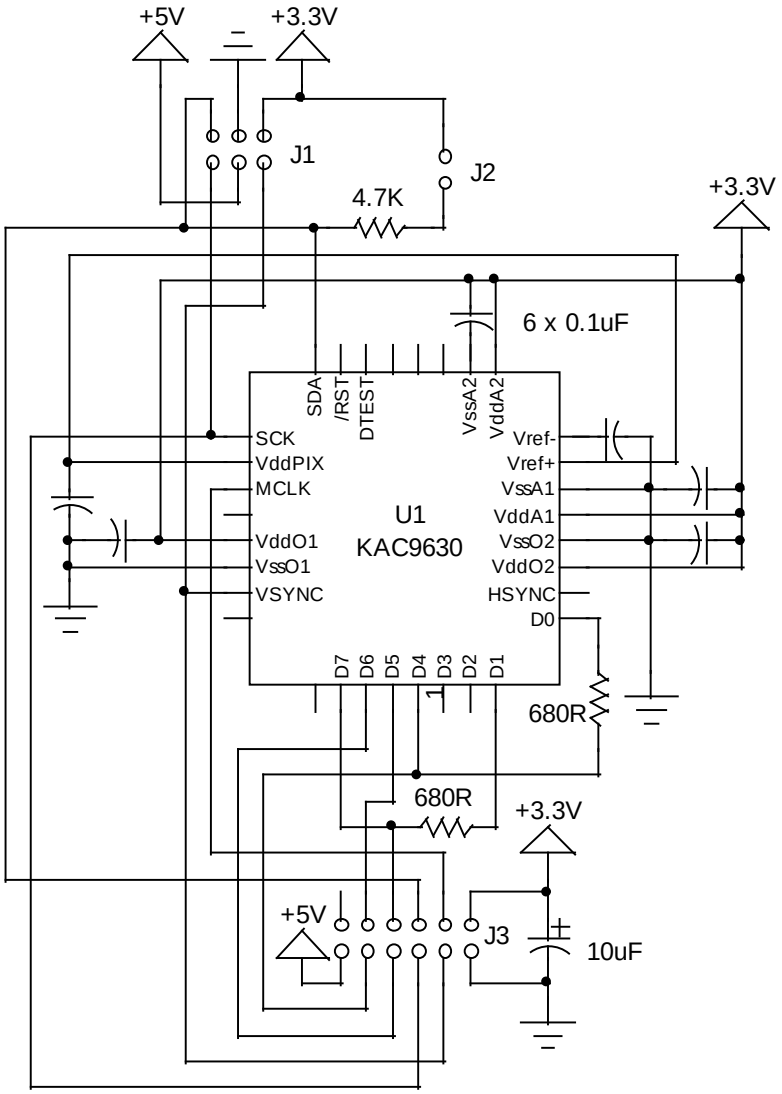

Figure 8: PropCAM-DB Schematic

Document Revision History

Version 1.0: Original release 10/28/2013
Version 1.1: Updated Features