Interfacing the C328 Camera with the University Program's RS232 UART

Xiang Gao and Kwan Yin Lau

Feb. 23, 2014

Introduction

The camera we are using in our project is the C328-7640 camera manufactured by COMedia Ltd. The camera uses RS232 UART to communicate with other component, which in our case is the Altera DE2 board. In our project we will be using GPIO to make connection with the camera since the working voltage of the camera is 3.3V. The basic operations of the camera are well explained in the user's manual. However, setting up the camera has been a challenge due to the limitation of University Program's RS232 UART implementation. Hopefully this app-note will ease the setup of not only the camera, but also anything else that is using this specific RS232 UART implementation.

Command Sets

As stated in the camera's user manual, each command is consisted of a six-byte instruction with a fixed prefix 0xAA. The commands themselves are straight-forward, but the University Program's RS232 UART only allows information to be transferred one byte at a time. Therefore, a small work-around is required to get things working.

Writing Commands

For example, to write a command to the camera is implemented as follows:

```
void C328_write_cmd(C328_COMMAND cmd) {
      alt_u32 write_FIFO_space;
      alt_u8 data_W8;

      data_W8 = cmd.Prefix;
      write_FIFO_space = alt_up_rs232_get_available_space_in_write_FIFO(
                  C328_dev);
      if (write_FIFO_space > WRITE_FIFO_EMPTY) {
            if (alt_up_rs232_write_data(C328_dev, data_W8) == 0)
                  alt_printf("%x ", data_W8);
      }

      data_W8 = cmd.ID_Number;
      write_FIFO_space = alt_up_rs232_get_available_space_in_write_FIFO(
                  C328_dev);
      if (write_FIFO_space > WRITE_FIFO_EMPTY) {
            if (alt_up_rs232_write_data(C328_dev, data_W8) == 0)
                  alt_printf("%x ", data_W8);
      }

      data_W8 = cmd.Parameter1;
      write_FIFO_space = alt_up_rs232_get_available_space_in_write_FIFO(
                  C328_dev);
      if (write_FIFO_space > WRITE_FIFO_EMPTY) {
            if (alt_up_rs232_write_data(C328_dev, data_W8) == 0)
                  alt_printf("%x ", data_W8);
      }

      data_W8 = cmd.Parameter2;
      write_FIFO_space = alt_up_rs232_get_available_space_in_write_FIFO(
                  C328_dev);
      if (write_FIFO_space > WRITE_FIFO_EMPTY) {
            if (alt_up_rs232_write_data(C328_dev, data_W8) == 0)
                  alt_printf("%x ", data_W8);
      }

      data_W8 = cmd.Parameter3;
      write_FIFO_space = alt_up_rs232_get_available_space_in_write_FIFO(
                  C328_dev);
      if (write_FIFO_space > WRITE_FIFO_EMPTY) {
            if (alt_up_rs232_write_data(C328_dev, data_W8) == 0)
                  alt_printf("%x ", data_W8);
      }

      data_W8 = cmd.Parameter4;
      write_FIFO_space = alt_up_rs232_get_available_space_in_write_FIFO(
                  C328_dev);
      if (write_FIFO_space > WRITE_FIFO_EMPTY) {
            if (alt_up_rs232_write_data(C328_dev, data_W8) == 0)
                  alt_printf("%x\n", data_W8);
      }

      OSTimeDlyHMSM(0, 0, 0, 10);
```

Note that a small delay is inserted at the end of function. Such delay is needed for proper transmission of the instructions.

Making Connection with the Camera

Once functions of writing and receiving commands are set, we can communicate with the camera. For example, to make connection with the camera, one can do the following:

```
int C328_connection() {
      alt_u32 sync_count = 1;
      alt_u32 read_FIFO_used;
      C328_COMMAND cmd_SYNC = C328_Set_SYNC();
      C328_COMMAND cmd_ACK = C328_Set_ACK(BLANK);
      C328_COMMAND ACK_from_camera;
      C328_COMMAND SYNC_from_camera;

      // Open the RS232 UART port
      C328_dev = alt_up_rs232_open_dev("/dev/rs232_0");
      if (C328_dev == NULL) {
            alt_printf("Error: could not open RS232 UART\n");
            return FAIL;
      } else
            alt_printf("Opened RS232 UART device\n");

      // Clear the read FIFO of the buffer
      C328_clear_read_FIFO();

      // Loop up to 60 times until ACK receive
      while (sync_count <= SYNCMAX) {
            // Write SYNC to the camera
            alt_printf("\nWrite SYNC to Camera attempt: %x\n", sync_count);
            C328_write_cmd(cmd_SYNC);

            // Wait for the camera's signal
            read_FIFO_used =
alt_up_rs232_get_used_space_in_read_FIFO(C328_dev);
            alt_printf("Char stored in read_FIFO: %x\n", read_FIFO_used);

            // Check if something is received
            if (read_FIFO_used == READ_FIFO_EMPTY) {
                  sync_count++;
            } else if (read_FIFO_used != READ_FIFO_EMPTY) {
                  alt_printf("Read ACK from Camera:\n");
                  C328_read_cmd(&ACK_from_camera);

                  alt_printf("Read SYNC from Camera:\n");
                  C328_read_cmd(&SYNC_from_camera);

                  // Check if ACK from camera is received
                  if ((ACK_from_camera.Prefix == PREFIX)
                              && (ACK_from_camera.ID_Number == ACK)
```

```
                              && (ACK_from_camera.Parameter1 == SYNC)) {

                        // Check if SYNC from camera is received
                        if ((SYNC_from_camera.Prefix == PREFIX)
                                    && (SYNC_from_camera.ID_Number == SYNC)) {
                              alt_printf("\nWrite ACK to Camera\n");
                              C328_write_cmd(cmd_ACK);
                              return SUCCESS;
                        }
                  }
            }
      }
      return FAIL;
}
```

Other commands can be implemented in similar way.

Note

Also attached is the driver class for the C328 camera by Shinichiro Nakamura. You may find it useful for setting up the camera.