

App and Microcontroller Info

Question: Can an App programmed on an electronic device be used as the invention?

The CIC Response: The goal of the CIC program is not the actual device, but the creative, observational and analytical processes the student goes through to create a functioning device. While it is possible to get a patent on software, the creation of an App is more aligned to the writing of a story than the physical creation of a working device. The writing of an App is strictly an intellectual activity and does not involve the physical, hands-on, real-world aspects and issues of dealing with real materials and forces that the CIC program is hoping the students will become aware of, address and overcome. However if the App is on a device that is then connected or interfaced to the real-world and does something physical, then that system is acceptable.

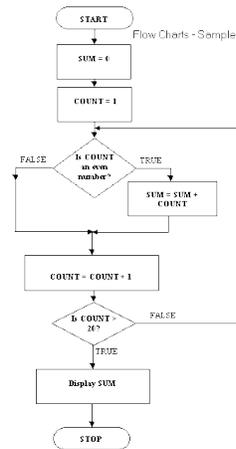
Examples of connecting to the real-world in a physical way are collecting temperature data, making sounds, responding to light, controlling motors, reading switches, sensing a magnetic field, turning on lights, etc.

The programmed App and the associated hardware to interface with the real-world could be connected to a full computer or even any of the many microcontroller boards such as made by Arduino, Parallax, Raspberry Pi, Intel or Micromint. Even individual microcontroller chips, such as those made by Microchip and others are fine.

However, beside the working invention system, the display must include the following additional information, so that the judges will be able to understand not only what the invention does, but also how it does it.

1. Logic Flow Chart

This a drawing showing all the logical steps that the App performs as it operates the device. The Flow Chart could be made with a CAD package or simply drawn by hand.



2. Code Printout

This is a listing of the computer code used to create the App. You can use any language you want for the App.

```

// OK when seek was successful
//
Z_OFF_T request = abs_offset + G.extra_bytes;
Z_OFF_T inbuf_offset = request % INBUFSIZ;
Z_OFF_T bufstart = request - inbuf_offset;

if (request < 0) {
  Info(slide, 1, ((char *)slide, LoadFarStringSmall(SeekMs
  G.zipfn, LoadFarString(ReportMsg)));
  return(FK_BADERR);
} else if (bufstart != G.cur_zipfile_bufstart) {
  Trace((stderr,
  "fpos_zip: abs_offset = %ld, G.extra_bytes = %ld\n",
  abs_offset, G.extra_bytes));
#ifdef USE_STRM_INPUT
  fseek(G.zipfd, bufstart, SEEK_SET);
  G.cur_zipfile_bufstart = ftell(G.zipfd);
#else /* !USE_STRM_INPUT */
  G.cur_zipfile_bufstart = fseek(G.zipfd, bufstart, SEEK
  
```

3. Electrical Schematic

This is a drawing of the electrical connections between the various parts showing how everything is connected together. This can be done with CAD software using the correct electronic symbols or simply draw rough images of the components by hand. You do NOT need to show the schematic of the Microcontroller itself, but just show the wires and components attached to the Microcontroller.

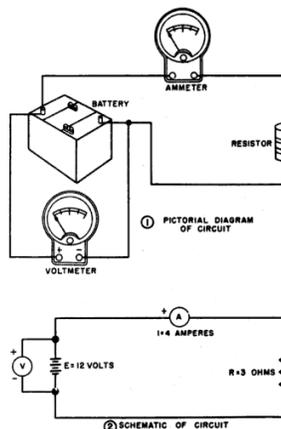


Figure 18. Diagram of a basic circuit.