**Embedded System Software프로젝트**

**Webcam control App & Device**

**(결과 보고서)**

**과목명: [CSE4116] 임베디드시스템소프트웨어**

**담당교수: 서강대학교 컴퓨터공학과 박 성 용**

**학번 및 이름: 120230200, 김지섭**

**개발기간: 2023. 06. 01. - 2023. 06. 26.**

**I. 개발 목표**

한 학기 동안 습득한 리눅스 시스템 프로그램 및 안드로이드 프로그래밍 기법을 이용해 실습 보드 상에서 자유 주제로 임베디드 소프트웨어를 개발한다.

**Project 목표:**

* Embedded Board에 USB webcam을 연결하여, Android App에 해당 영상을 송출한다.
* Webcam이 원하는 위치를 촬영할 수 있도록 방향을 조절할 수 있도록 한다.
* 사진 촬영 및 저장 후 Android의 갤러리에서 해당 이미지를 확인할 수 있도록 한다.

**II. 개발 범위 및 내용**

- 자신들이 설계한 개발 목표를 달성하기 위하여 어떠한 내용의 개발을 수행할 지 그 범위와 개발 내용을 기술할 것.

**가. 개발 범위**

Timer Device Driver 및 Test용 Example Application 개발

1. Example Application

실행 예시



1. Stopwatch module

* fpga\_fnd에 현재 측정된 시간을 출력한다.

\* fnd부분의 앞의 두 자리는 분(60 분),뒤의 두 자리는 초(60 초)를 표시한다.

\* Reset, Stop 시, fpga\_led 의 불을 꺼준다.(0000)

\* 키 입력은 Interrupt를 사용하여 수행한다.

* Home 버튼 -> start

\*1초마다 fnd의 정보를 갱신한다.(timer 사용)

* Back 버튼 -> pause

\* 일시 정지

* Vol+ 버튼 -> reset

\* fnd 출력 값 및 시간이 모두 초기상태로 돌아간다

* Vol- 버튼 -> stop

\* 3초 이상 누르고 있을 시, application 종료

\* fnd를 0000으로 초기화 한다.

**나. 개발 내용**

1. Flow Chart

Application and module

텍스트, 도표, 라인, 평면도이(가) 표시된 사진

자동 생성된 설명

Buttom half Logic

도표, 스케치, 평면도, 기술 도면이(가) 표시된 사진

자동 생성된 설명

1. Android Application 구현

|  |
| --- |
| public class MainActivity extends Activity implements OnTouchListener, OnClickListener {  private static final String TAG = "WebCam";  private native int moveMotorRight();  private native int moveMotorLeft();  private native int stopMotor();  private native byte[] capture();  ImageView webCam;  Bitmap bitmap = null;  byte[] source = null;  Thread thread;  static{  System.loadLibrary("motor\_jni");  };  } |

|  |
| --- |
| @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  Log.d(TAG, "Start Application");  setContentView(R.layout.fragment\_main);  View leftButton = findViewById(R.id.left\_button);  leftButton.setOnTouchListener(this);  View rightButton = findViewById(R.id.right\_button);  rightButton.setOnTouchListener(this);  View shootButton = findViewById(R.id.shoot\_button);  shootButton.setOnClickListener(this);  webCam = (ImageView) findViewById(R.id.web\_cam);    if(thread == null) {  thread = new Thread(new WebCamImage());  }  thread.start();    } |

|  |
| --- |
| @Override  public boolean onTouch(View v, MotionEvent event) {  // TODO Auto-generated method stub  switch(v.getId()) {  case R.id.right\_button :  switch(event.getAction()) {  case MotionEvent.ACTION\_DOWN:  // PRESSED  Log.d(TAG, "Pressed Right");  Log.d(TAG, String.valueOf(moveMotorRight()));  return true; // if you want to handle the touch event  case MotionEvent.ACTION\_UP:  // RELEASED  Log.d(TAG, "Released Right");  Log.d(TAG, String.valueOf(stopMotor()));  return true; // if you want to handle the touch event  }  case R.id.left\_button :  switch(event.getAction()) {  case MotionEvent.ACTION\_DOWN:  // PRESSED  Log.d(TAG, "Pressed Left");  Log.d(TAG, String.valueOf(moveMotorLeft()));  return true; // if you want to handle the touch event  case MotionEvent.ACTION\_UP:  // RELEASED  Log.d(TAG, "Released Left");  Log.d(TAG, String.valueOf(stopMotor()));  return true; // if you want to handle the touch event  }  }  return false;  } |

|  |
| --- |
| public void onClick(View v) {  // TODO Auto-generated method stub  switch(v.getId()){  case R.id.shoot\_button:  SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd-hh-mm-ss");  Log.d(TAG, formatter.format(new Date()));  OutputStream out = null;  try {  File file = new File("/sdcard/DCIM", formatter.format(new Date()) + ".png");  file.createNewFile();  out = new FileOutputStream(file);  Context c = this.getBaseContext();  c.sendBroadcast(new Intent(Intent.ACTION\_MEDIA\_SCANNER\_SCAN\_FILE, Uri.fromFile(file)));  bitmap.compress(Bitmap.CompressFormat.PNG, 80, out);  } catch (FileNotFoundException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (IOException e) {  // TODO Auto-generated catch block  e.printStackTrace();  }  }  } |

|  |
| --- |
| class WebCamImage implements Runnable{  Handler handler = new Handler();  boolean hasImageAlready = (webCam.getDrawable() != null);  @Override  public void run() {  // TODO Auto-generated method stub  while(true) {  // get Image  Log.d(TAG, "Get Image");  source = capture();  if(source != null) {  Log.d(TAG, "Image Not NULL : ");  // decode Image  ByteArrayOutputStream out = new ByteArrayOutputStream();  YuvImage yuvImage = new YuvImage(source, ImageFormat.YUY2, 640, 480, null);  yuvImage.compressToJpeg(new Rect(0, 0, 480, 360), 100, out);  byte[] imageBytes = out.toByteArray();    bitmap = BitmapFactory.decodeByteArray(imageBytes, 0, imageBytes.length);  Log.d(TAG, bitmap.getConfig().name());    if(bitmap != null) {  Log.d(TAG, "Bitmap Not NULL");  runOnUiThread(new Runnable() {  @Override  public void run() {  // TODO Auto-generated method stub  webCam.setImageBitmap(bitmap);  Log.d(TAG, "Set Image");  }  });  }  }  }  }  } |

1. Stopwatch module구현

**III. 추진 일정 및 개발 방법**

1. **추진 일정**

**텍스트, 번호, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명**

1. **개발 방법**

개발해야 하는 기능들을 button단위로 구분하여 waterfall 방식으로 개발하였습니다.

우선 기존 실습 코드를 참고하여, module을 실행시킬 수 있는 test application을 구현하였습니다.

이후 module을 구현하였습니다. 이때, button 별로 수행할 기능을 각각 순차적으로 구현하였습니다.

**IV. 연구 결과**

 Interrupt를 활용하여, process가 sleep 상태인 경우에도 button입력과 관련된 동작을 수행할 수 있었습니다. 특히 bottom half로직이 수행된 이후, 다시 tasklet을 다시 setting하여, tasklet을 반복 수행하도록 사용할 수 있다는 것을 알게 되었습니다. 마지막으로 wake up을 위한 trigger를 설정하여, process를 깨울 수 있었습니다.

**V. 기타**

Interuupt와 Time를 활용하여, 직접 개발해보는 경험을 통해, 강의에 대한 이해도가 높아진 것 같습니다. Top half와 Bottom half를 적절하게 사용하기 위해 고민하면서, 좀 더 확실하게 개념을 잡을 수 있었던 것 같습니다.