Modular Programming:
Where the rubber meets the Code

Gabriel Hugh Elkaim
Announcements

1. One study must remain intact.

2. Beware of shorts — Binky is sick.

3. Lower center —

4. AAskews — run job.
Summarizing (cont’d):

- Replaced blower on the large cutter.
- Bean challenge - Tuesday ~ 6pm.

3 out of 3 successful.
The three uses of **static**

(1) Allocates a variable at startup & retains its value between function calls. = \&. Pervasive

(2) Scope restriction — cannot be accessed from outside the module

static int foo (void);

Public/Private

(3) Pseudo-global variable — Module external variable

static uint64_t foo-inner;

uint64_t Read-foo-inner (void) { return foo-inner; }
Encapsulation (1.2)

int foo

char bar

FunctionA()

FunctionB()

FunctionC()

Sending

Data

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Implemented

Impl
Encapsulation (2.2)

- Interface definition "wrapped"

- Implement std functionality

    \begin{align*}
    & \text{RC_Init/RC_End} \\
    & \text{RC_AddPin/RC_RemovePin} \\
    & \text{RC_SetPin/RC_GetPin}
    \end{align*}

- Module: 'name.h'
- Source: 'name.c'

```c
int foo
char bar

FunctionA()
FunctionB()
FunctionC()
```
What goes into a header file?

```cpp
#include guard

// Blurb comment - explains module

prototypes for public functions

#define for public constants

documentation comments /*

Public data structures { }
```
What shouldn't be in a header file?

Not going in .h file:

- No Functions
- No EXECUTABLE CODE
- No Module VARIABLES
- No STATIC Function PROTOTYPES in .c

.c - implementation - PRIVATE

.h - interface - PUBLIC
Where do you #include the header file?

Everywhere I might need access to its

 vaccinated

```c
#include _motordrive_H
#define _motordrive_H

// code
#endif
```
Programming Style Issues

Layout within a module

K & R

**NOT SWIFT: “F”**

One "True" Indentation Style

```c
if (NewTime() && ((GetTime() % 1050) == 0))
  return 1;
else
  return 0;
```

```c
void putKey(SERVICE_PARAM)
{
  putchar(GET_SHARED_BYTE());
  putchar('.');
}
```

Use of White Space

Use of Comments

Module Level Comment

```c
#include

library/module headers
Public header for this module
Private Constant Definitions
Private Macro Definitions
Private Type Definitions
Private Variables
Private Function Prototypes
Code
```

Test Unness

while (1)
{
  MSE_HandleEvents();
}

(-14 * p -> head s s w)

count++; // next and

Naming Conventions

Gabriel Hugh Elkaim
Module Design by Interface Specification

• **View**
  – The module provides **Services** to the rest of the code

• **Design Activities**
  – Specify the services
    • **Describe Functionality**
  – **Name the Services**
  – Design the implementation
Application Programming Interfaces (API) for the ME218c Master modules

A Real Example

Module: Communications to UI on PC

To avoid hanging up the master during the transmission or reception of messages, this module should implement buffered, interrupt driven transmit and receive. The communications routines for this module will need to be interrupt driven because the UI may send its message at any time.

Char InitializeUICommunications (void);
Do whatever hardware and software initialization necessary to prepare for communications with the UI on SC1.

Void TellUINewUserReady (void);
Should send the message to the UI that a new iButton has been inserted and read.

Unsigned char IsNameReady (void);
Should check to see if a new name is ready from the UI. Return TRUE if a new name is ready, FALSE otherwise.

Unsigned char GetNewName (unsigned char NameSpace[]);
Should copy the name gotten from the UI into the array NameSpace. The copy operation should copy no more than 16 characters, including the terminating NULL. Should return TRUE if there was a new name ready, FALSE otherwise.
Design the interfaces to modules

• Design interface for:
  – Driving the platform
  – Gathering Sensor data

• Produce
  – Public Interface specification
  – What are the details that are being hidden?
uint8_t DriveInit(void) - set up hardware/software to drive my motors, return SUCCESS or FAILURE

DriveFullStop(void) - stop all motors
DriveStraight(int8_t speed) - drive both motors at the same speed, range from -100 to 100, negative values indicate reverse.

DriveTurn(enum type, int8_t speed) - turns the robot, speed range from -100 to +100, negative indicates CCW rotation, type of turns are: tank, pivot, arc, slow

Drive End(void)
BAT_VOLTAGE 287

\[ \frac{287}{1023} \times 33 = 9.258 \text{ V} \]

\[ \frac{7}{1} - 0.6 \text{ V} = 8.65 \text{ V} \]

\[ \frac{6}{8.65} = 0.6929 \rightarrow 69\% \text{ duty cycle} \]
STATE EXPLOSION PROBLEM

Hierarchical State Machines

- Harel's Statecharts

Allows us to zoom in and out based on behavior.
A Possible Top-Level State Diagram

- **Enter**
  - Finding Tape I
  - Found Tape
  - Tracking Tape
  - Tracking Tape
  - Finding Tape II
  - Found Tape
  - Found Tape
  - Stop
Work out State Diagrams to Implement Finding Tape I

Diagram:
- Start state: Beacon
- States:
  - Going Forward
  - Hit Tape
- Transitions:
  - Right, Stay
  - Very Slow, Stay
  - Both, Go Back
  - Post, Tape Found

Gabriel Hugh Elkaim
Announcement

1. Field is up in BES-138
2. Keep it pristine
3. No coding in BES-138
4. $4/meal
5. Beer — Tuesday 22/Nov

Roboslaugh
3 at 9:30
3 balls into 2 targets
Acquiring

Entry → Turning CW → Left Sensor Peaks → Set Slow Speed, Reverse → Turning CCW → L ≈ R → Stop Turning, Post Beacon Acquired → Acquired
Implementing Hierarchical State Machines

- What do you need?

Fall down to state values
↓
pass events down/back

receive call for left and read
on edge of state.
1 \rightarrow 2

\begin{align*}
2 & \rightarrow \text{entry} \\
2 & \rightarrow \text{exit}
\end{align*}
State Machine Function Template

If current state is state one
  Execute During function for state one
  If an event is active
    If event is event one
      Execute action function for state one: event one
      Decide what the next state will be
    Endif
  If event is event two
    Execute action function for state one: event two
    Decide what the next state will be
  Endif
Repeat the block above as required for each of the possible events affecting this state.
If next state is different from current state
  Execute exit function for state one
  Execute entry function for new state
  Modify state variable to reflect the new state
Endif
Endif
Return from state machine function
Module
d:\me218b\Lectures\Lecture 29\SMTemplate.c

Description
This is a template file for implementing state machines.

Notes

History
When      Who      What/Why
----------- --- -------
02/18/99  10:19 jec  built template from MasterMachine.c
02/14/99  10:34 jec  Began Coding

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#define NO_OF_STATES 5
#define NO_OF_EVENTS 3

typedef enum {STATE_1, STATE_2, STATE_3, STATE_4, STATE_5} State;
typedef enum {EVENT_1, EVENT_2, EVENT_3} Event;

State CurrentState;

void Entry(void);
void Exit(void);

void State1(void);
void State2(void);
void State3(void);
void State4(void);
void State5(void);

int main(void)
{
    int i;
    for (i = 0; i < NO_OF_EVENTS; i++)
        Entry();
    CurrentState = STATE_1;
    for (i = 0; i < NO_OF_EVENTS; i++)
        State1();
    CurrentState = STATE_2;
    for (i = 0; i < NO_OF_EVENTS; i++)
        State2();
    CurrentState = STATE_3;
    for (i = 0; i < NO_OF_EVENTS; i++)
        State3();
    CurrentState = STATE_4;
    for (i = 0; i < NO_OF_EVENTS; i++)
        State4();
    CurrentState = STATE_5;
    for (i = 0; i < NO_OF_EVENTS; i++)
        State5();
    Exit();
    return 0;
}

void Entry(void)
{
    // Entry functions
}

void Exit(void)
{
    // Exit functions
}

void State1(void)
{
    // State 1 functions
}

void State2(void)
{
    // State 2 functions
}

void State3(void)
{
    // State 3 functions
}

void State4(void)
{
    // State 4 functions
}

void State5(void)
{
    // State 5 functions
}
void RunStateMachine(unsigned char CurrentEvent)
{
    unsigned char NextState = CurrentState;

    switch (CurrentState)
    {
    case STATE_ONE:  // If current state is state one
        DuringStateOne(); // Execute During function for state one
        if (CurrentEvent != NO_EVENT) // If an event is active
        {
            switch (CurrentEvent)
            {
                case EVENT_ONE: // If event is event one
                    // Execute action function for state one: event one
                    NextState = STATE_TWO; // Decide what the next state will be
                    break;
            }
        }
        // If next state is different from current state
        if (NextState != CurrentState)
        {
            // Execute exit function for current state
            // Execute entry function for new state
            CurrentState = NextState; // Modify state variable
        }
    return;
}
/*************/

Function

StartStateMachine

Parameters

None

Returns

None

Description

Does any required initialization for this state machine

Notes

Author

J. Edward Carryer, 2/18/99, 10:38AM

void StartStateMachine ( void )
{
    CurrentState = ENTRY_STATE;
    // call the (optional) entry function for the ENTRY_STATE
    // any other initialization necessary to re-start the state machine
}
Questions?
Noise Isolation: Keeping the Gremlins Out!

Gabriel Hugh Elkaim
How noise gets into your circuits

Figure 1-11. Before noise can be a problem, there must be a noise source, a receptor that is susceptible to the noise, and a coupling channel that transmits the noise to the receptor.
Key Characteristics of the **noise source**

- **Voltage**
  - high voltage \( \frac{dv}{dt} \) \( \rightarrow \) \( I = C \frac{dv}{dt} \)
  - strong electric field - capacitive coupling

- **Current**
  - high current \( \frac{di}{dt} \) - strong magnetic field
  - inductive coupling

- **Frequency**
  - high frequency - radiation, radiation耦合
  - \( 0 \leq \text{distance} \leq \text{wavelength} \)

- **Distance from the victim**
  - \( \phi \) DIRECT CONTACT
  - Conductor coupling
What is the most likely coupling mechanism for:

- Florescent Light noise - 100's of volts, capacitive coupling
- Arc Welding Noise - High current (100's of amps) in arc phase
- Digital Clock Noise
Conductive coupling (1.2)
Conductive coupling (2.2)

+12V

CIRCUIT 1

GROUND VOLTAGE CIRCUIT 1

GROUND CURRENT 1

COMMON GROUND IMPEDANCE

GROUND CURRENT 2

GROUND VOLTAGE CIRCUIT 2

+5V
Avoid ground loops

How should I wire these up?

(1) Low power path to things

12V

5V

Motor

Sensor

PIC 182
Which waveform **must** be conductively coupled?

Why?

**Because it has DC component.**
Identifying Characteristics of Conductive Coupling

(1) Metallic contact is required

(2) Unshielded by cable movement or people

(3) Non-zero DC component on its waveform

Break the conductive contact
- Route wires (power/ground) back to +114
- Use a timer fitter 0-1000
Capacitively coupled noise

Coupling capacitance

Simplified circuit
Physical Representation of capacitively coupled noise

**Diagram:**
- **Signal:** Connections from conductors 1 to 2.
- **Noise Source:** Input voltage $V_1$.
- **Noise:** Capacitances $C_{16}$ and $C_{20}$.
- **Unmodeled:** Various components and connections.
- **Physical Representation:** A diagram illustrating the noise and signal paths.

**Annotations:**
- "I can control this" on the physical representation.
- "Can I do anything about this?" on the signal path.
Equivalent circuit for capacitively coupled noise

\[ V_N = \frac{j\omega[C_{12}/(C_{12} + C_{2G})]}{j\omega + 1/R(C_{12} + C_{2G})} V_1 \]

If \( R \gg \frac{1}{j\omega(C_{12} + C_{2G})} \)

\[ V_N = \frac{C_{12}}{(C_{12} + C_{2G})} V_1 \]

If \( R \ll \frac{1}{j\omega(C_{12} + C_{2G})} \)

\[ V_N = j\omega R C_{12} V_1 \]
Reducing Capacitively Coupled Noise

Position my shield to intercept the noise.
Summary of capacitive noise reduction techniques

1. Reduce capacitive coupling
2. Use a shield, ground only one side
3. Reduce circuit impedance
Isolation

Split my circuits so there are no common paths.
Isolation via magnetic coupling

RELAY

PWM

PWM
PCB mount miniature relays
Reed Relay construction

Axial leads allow various mounting configurations.

Nickel plated steel jacket provides magnetic shields.

Coil windings on reed switch for increased electrical efficiencies.

Ends epoxy sealed to provide environmental protection.

End flange for coil winding retainer.
Optical Isolation

[Diagram of optical isolation circuit with components labeled: +5, 4N36, 330Ω, 2.2k, HC(T), AC(T), TTL, V+]
## General Purpose 6-Pin Phototransistor Optocouplers

<table>
<thead>
<tr>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>4N25</td>
</tr>
<tr>
<td>4N26</td>
</tr>
<tr>
<td>4N27</td>
</tr>
<tr>
<td>4N28</td>
</tr>
<tr>
<td>4N35</td>
</tr>
<tr>
<td>4N36</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>H11A1</td>
</tr>
<tr>
<td>H11A2</td>
</tr>
<tr>
<td>H11A3</td>
</tr>
<tr>
<td>H11A4</td>
</tr>
<tr>
<td>H11A5</td>
</tr>
</tbody>
</table>

### White Package (-M Suffix)

![White Package Diagram](image)

### Black Package (No -M Suffix)

![Black Package Diagram](image)

### Schematic

![Schematic Diagram](image)

*PIN 1: ANODE
2: CATHODE
3: NO CONNECTION
4: Emitter
5: Collector
6: BASE*
### Electrical Characteristics

\( T_A = 25^\circ C \) unless otherwise specified.

### Individual Component Characteristics

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Test Conditions</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ*</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Emitter</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Forward Voltage</td>
<td>( I_F = 10 \text{ mA} )</td>
<td>( V_F )</td>
<td>1.18</td>
<td>1.50</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Reverse Leakage Current</td>
<td>( V_R = 6.0 \text{ V} )</td>
<td>( I_R )</td>
<td>0.001</td>
<td>10</td>
<td></td>
<td>( \mu \text{A} )</td>
</tr>
<tr>
<td><strong>Detector</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collector-Emitter Breakdown Voltage</td>
<td>( I_C = 1.0 \text{ mA}, I_F = 0 )</td>
<td>( \text{BV}_{CEO} )</td>
<td>30</td>
<td>100</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Collector-Base Breakdown Voltage</td>
<td>( I_C = 100 \mu \text{A}, I_F = 0 )</td>
<td>( \text{BV}_{CBO} )</td>
<td>70</td>
<td>120</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Emitter-Collector Breakdown Voltage</td>
<td>( I_E = 100 \mu \text{A}, I_F = 0 )</td>
<td>( \text{BV}_{EO} )</td>
<td>7</td>
<td>10</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Collector-Emitter Dark Current</td>
<td>( V_{CE} = 10 \text{ V}, I_F = 0 )</td>
<td>( I_{CEO} )</td>
<td>1</td>
<td>50</td>
<td></td>
<td>nA</td>
</tr>
<tr>
<td>Collector-Base Dark Current</td>
<td>( V_{CB} = 10 \text{ V} )</td>
<td>( I_{CBO} )</td>
<td></td>
<td>20</td>
<td></td>
<td>nA</td>
</tr>
<tr>
<td>Capacitance</td>
<td>( V_{CE} = 0 \text{ V}, f = 1 \text{ MHz} )</td>
<td>( C_{CE} )</td>
<td></td>
<td>8</td>
<td></td>
<td>( \text{pF} )</td>
</tr>
<tr>
<td>DC Characteristic</td>
<td>Test Conditions</td>
<td>Symbol</td>
<td>Device</td>
<td>Min</td>
<td>Typ*</td>
<td>Max</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4N35</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4N36</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4N37</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H11A1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H11A5</td>
<td>50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Current Transfer Ratio, Collector to Emitter</td>
<td>(I_F = 10 mA, V_CE = 10 V)</td>
<td>CTR</td>
<td>4N25</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4N26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H11A2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H11A3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4N27</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4N28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H11A4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4N35</td>
<td></td>
<td>40</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4N36</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4N37</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(I_F = 10 mA, V_CE = 10 V, T_A = -55°C)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(I_F = 10 mA, V_CE = 10 V, T_A = +100°C)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collector-Emitter Saturation Voltage</td>
<td>((I_C = 2 \text{ mA}, I_F = 50 \text{ mA}))</td>
<td>(V_{CE\ (SAT)})</td>
<td></td>
<td>(0.5)</td>
<td>(V)</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>---------------------------------</td>
<td>-----------------</td>
<td>---</td>
<td>-------------------</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>((I_C = 0.5 \text{ mA}, I_F = 10 \text{ mA}))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC Characteristic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-Saturated Turn-on Time</td>
<td>((I_F = 10 \text{ mA}, V_{CC} = 10 \text{ V}, R_L = 100\Omega)) ((\text{Fig.20}))</td>
<td>(T_{ON})</td>
<td>4N25</td>
<td>4N26</td>
<td>4N27</td>
<td>4N28</td>
</tr>
<tr>
<td>Non Saturated Turn-on Time</td>
<td>((I_C = 2 \text{ mA}, V_{CC} = 10 \text{ V}, R_L = 100\Omega)) ((\text{Fig.20}))</td>
<td>(T_{ON})</td>
<td>4N35</td>
<td>4N36</td>
<td>4N37</td>
<td></td>
</tr>
</tbody>
</table>
AC INPUT/PHOTOTRANSISTOR OPTOCOUPLERS

**DESCRIPTION**
The H11AAX series consists of two gallium-arsenide infrared emitting diodes connected in inverse parallel driving a single silicon phototransistor output.

**FEATURES**
- Bi-polar emitter input
- Built-in reverse polarity input protection
- Underwriters Laboratory (UL) recognized — File #E90700
- VDE approved — File #E94766 (ordering option ‘300’)

**APPLICATIONS**
- AC line monitor
- Unknown polarity DC sensor
- Telephone line interface
PHOTODARLINGTON OPTOCOUPLERS

DESCRIPTION
The CNX48U, H11B1, H11B2, H11B255, H11B3, MOC8080, TIL113 have a gallium arsenide infrared emitter optically coupled to a silicon planar photodarlington.

FEATURES
• High sensitivity to low input drive current
• Meets or exceeds all JEDEC Registered Specifications
• VDE 0884 approval available as a test option
  -add option .300. (e.g., H11B1.300)

APPLICATIONS
• Low power logic circuits
• Telecommunications equipment
• Portable electronics
• Solid state relays
• Interfacing coupling systems of different potentials and impedances.
<table>
<thead>
<tr>
<th>AC Characteristics</th>
<th>$t_{on}$</th>
<th>$t_{off}$</th>
<th>$t_{on}$</th>
<th>$t_{off}$</th>
<th>$t_{on}$</th>
<th>$t_{off}$</th>
<th>$t_{on}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(I_C = 10 \text{ mA}, V_{CE} = 10 \text{ V})$</td>
<td>H11B1</td>
<td>25</td>
<td>H11B2</td>
<td>36</td>
<td>H11B255</td>
<td>70</td>
<td>H11B3</td>
</tr>
<tr>
<td>$(R_L = 100 \Omega)$ (Fig. 7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(I_F = 10 \text{ mA}, V_{CC} = 5 \text{ V})$</td>
<td>CNX48U</td>
<td>70</td>
<td></td>
<td></td>
<td>MOC8080</td>
<td>190</td>
<td></td>
</tr>
<tr>
<td>$(R_E = 100 \Omega), (R_{BE} = 1 \text{ M}\Omega)$ (Fig. 8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(I_F = 1 \text{ mA}, V_{CC} = 5 \text{ V})$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(R_E = 1 \text{k}\Omega), (R_{BE} = 10 \text{ M}\Omega)$ (Fig. 8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(I_F = 5 \text{ mA}, V_{CC} = 10 \text{ V})$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(R_L = 100 \Omega)$ (Fig. 7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(I_F = 200 \text{ mA}, I_C = 50 \text{ mA})$</td>
<td>TIL113</td>
<td>0.35</td>
<td></td>
<td></td>
<td>TIL113</td>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td>$(V_{CC} = 10 \text{ V}) (R_L = 100 \Omega)$ (Fig. 7)</td>
<td></td>
<td>55</td>
<td></td>
<td></td>
<td></td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

**Switching Times**
6-Pin DIP Optoisolators
Logic Output

The H11L1 and H11L2 have a gallium arsenide IRED optically coupled to a high-speed integrated detector with Schmitt trigger output. Designed for applications requiring electrical isolation, fast response time, noise immunity and digital logic compatibility.

- Guaranteed Switching Times — $t_{on}$, $t_{off} < 4 \mu s$
- Built-In On/Off Threshold Hysteresis
- High Data Rate, 1 MHz Typical (NRZ)
- Wide Supply Voltage Capability
- Microprocessor Compatible Drive
- To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option.

Applications
- Interfacing Computer Terminals to Peripheral Equipment
- Digital Control of Power Supplies
- Line Receiver — Eliminates Noise
- Digital Control of Motors and Other Servo Machine Applications
- Logic to Logic Isolator
- Logic Level Shifter — Couples TTL to CMOS

MAXIMUM RATINGS ($T_A = 25^\circ C$ unless otherwise noted)
## H11L1 H11L2

**Electrical Characteristics** *(Ta = 25°C unless otherwise noted)*

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ(1)</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input LED</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reverse Leakage Current <em>(VR = 3 V, RL = 1 MΩ)</em></td>
<td>Iᵣ</td>
<td>—</td>
<td>0.05</td>
<td>10</td>
<td>μA</td>
</tr>
<tr>
<td>Forward Voltage <em>(IF = 10 mA)</em></td>
<td>Vᵥ</td>
<td>—</td>
<td>1.2</td>
<td>1.5</td>
<td>Volts</td>
</tr>
<tr>
<td><em>(IF = 0.3 mA)</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capacitance <em>(VR = 0 V, f = 1 MHz)</em></td>
<td>C</td>
<td>—</td>
<td>18</td>
<td>—</td>
<td>pF</td>
</tr>
<tr>
<td><strong>Output Detector</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating Voltage</td>
<td>Vcc</td>
<td>3</td>
<td>—</td>
<td>15</td>
<td>Volts</td>
</tr>
<tr>
<td>Supply Current <em>(IF = 0, VCC = 5 V)</em></td>
<td>Icc(on)</td>
<td>—</td>
<td>1.6</td>
<td>5</td>
<td>mA</td>
</tr>
<tr>
<td>Output Current, High <em>(IF = 0, VCC = Vo = 15 V)</em></td>
<td>IOH</td>
<td>—</td>
<td>—</td>
<td>100</td>
<td>μA</td>
</tr>
<tr>
<td><strong>Coupled</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supply Current <em>(IF = IF(on), VCC = 5 V)</em></td>
<td>Icc(on)</td>
<td>—</td>
<td>1.6</td>
<td>5</td>
<td>mA</td>
</tr>
<tr>
<td>Output Voltage, Low <em>(RL = 270 Ω, VCC = 5 V, IF = IF(on)</em></td>
<td>Vol</td>
<td>—</td>
<td>0.2</td>
<td>0.4</td>
<td>Volts</td>
</tr>
<tr>
<td>Threshold Current, ON <em>(RL = 270 Ω, VCC = 5 V)</em></td>
<td>H11L1</td>
<td>—</td>
<td>1.2</td>
<td>1.6</td>
<td>mA</td>
</tr>
<tr>
<td><em>(H11L2)</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Threshold Current, OFF <em>(RL = 270 Ω, VCC = 5 V)</em></td>
<td>H11L1</td>
<td>—</td>
<td>0.3</td>
<td>0.75</td>
<td>mA</td>
</tr>
<tr>
<td><em>(H11L2)</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hysteresis Ratio <em>(RL = 270 Ω, VCC = 5 V)</em></td>
<td>Iff</td>
<td>—</td>
<td>0.5</td>
<td>0.75</td>
<td>—</td>
</tr>
<tr>
<td><em>(Iff)</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Isolation Voltage <em>(2)</em> 60 Hz, AC Peak, 1 second, Ta = 25°C</td>
<td>VISO</td>
<td>7500</td>
<td>—</td>
<td>—</td>
<td>Vac(pk)</td>
</tr>
<tr>
<td>Turn–On Time <em>(RL = 270 Ω)</em></td>
<td></td>
<td></td>
<td>1.2</td>
<td>4</td>
<td>μs</td>
</tr>
<tr>
<td>Fall Time <em>(VCC = 5 V, IF = IF(on)</em></td>
<td></td>
<td>—</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Turn–Off Time <em>(Ta = 25°C)</em></td>
<td></td>
<td></td>
<td>1.2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Rise Time <em>(Rᵣ)</em></td>
<td></td>
<td>—</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Always design to the specified minimum/maximum electrical limits (where applicable).
2. For this test, IRED Pins 1 and 2 are common and Output Gate Pins 4, 5, 6 are common.
3. RL value effect on switching time is negligible.
Create a design to connect

3.3V

UNO 32

PS2501
Opto-Isolator

LM298
H-Bridge

V_{in}

Gabriel Hugh Elkaim

CMPE 118/218 – Intro. to Mechatronics
Galvanic Isolation
Questions?
T-shirts  -  Plazza
Timers and Analog to Digital Conversion: Tick tock tick tock...

Gabriel Hugh Elkaim
Interfacing to the Real World

The world

Sensors

MC

ADC
I/O Ports
Timers
Comm

Actuators

PWM
I/O Ports
Comm
D/A

CMPE 118/218 – Intro. to Mechatronics
Timers

String of Flip Flops

counter which is driven by a fixed time base.
Controller - median complexity

283 - 325 ft
995 - 32 ft

Dedicated chips (GPS)

RTCC ~ 1 sec/year

GPS - PPS > 2

< 1 sec/year
Figure 16-1: Output Compare Module Block Diagram

- Set Flag bit OCxIF
- Output Logic
- Output Enable
- OCTSEL
- TMR register inputs from time bases
- Period match signals from time bases
- OCFA or OCFB
- OCM<2:0>-Mode Select

Symbolic notation:
- OCxRS
- OCxR
- Comparator
- OCTSEL
- 16
- 16
Figure 15-1: Input Capture Module Block Diagram
Figure 17.1: 10-bit High-Speed ADC Block Diagram

Note 1: $V_{REF+}$ and $V_{REF-}$ inputs can be multiplexed with other analog inputs.
Figure 17-5: Simplified 10-bit High-Speed ADC Block Diagram for Alternate Sample Mode
**CONVERSION CODE**

<table>
<thead>
<tr>
<th>RANGE OF ANALOG INPUT VALUES</th>
<th>DIGITAL OUTPUT CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>45 - 55</td>
<td>0 ... 101</td>
</tr>
<tr>
<td>35 - 45</td>
<td>0 ... 100</td>
</tr>
<tr>
<td>25 - 35</td>
<td>0 ... 011</td>
</tr>
<tr>
<td>15 - 25</td>
<td>0 ... 010</td>
</tr>
<tr>
<td>0.5 - 1.5</td>
<td>0 ... 001</td>
</tr>
<tr>
<td>0 - 0.5</td>
<td>0 ... 000</td>
</tr>
</tbody>
</table>

Ideal Straight Line

Step Width (1 LSB)

Midstep Value of 0 ... 011

Inherent Quantization Error (± 1/2 LSB)

Elements of Transfer Diagram for an Ideal Linear ADC
Comparador

Nota 1: En dispositivos con un módulo USB, y cuando el módulo está habilitado, esta señal está controlada por el módulo USB, por lo que no está disponible como una señal de entrada para el comparador.

Nota 2: Conectado internamente.
Comparator (2.2)
Figure 20.1: Comparator Voltage Reference Block Diagram

Note 1: These bits are not available on all devices. On such devices, CVREF is generated by the resistor network and IVREF is connected to 1.2V. Refer to the specific device data sheet for availability.
Elements of Transfer Diagram for an Ideal Linear DAC

<table>
<thead>
<tr>
<th>Digital Input Code</th>
<th>0...000</th>
<th>0...001</th>
<th>0...010</th>
<th>0...011</th>
<th>0...100</th>
<th>0...101</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog Output Value</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
Offset error of a Linear 3-Bit Natural Binary Code Converter (Specified at Step 000)

(a) ADC
(b) DAC
Gain Error of a Linear 3-Bit Natural Binary Code Converter (Specified at Step 111), After Correction of the Offset Error
End-Point Linearity Error of a Linear 3-Bit Natural Binary-Coded ADC or DAC
(Offset Error and Gain Error are Adjusted to the Value Zero)
Differential Linearity Error of a Linear ADC or DAC
Absolute Accuracy or Total Error of a Linear ADC or DAC

(a) ADC

Total Error At Step 0...101
(± 1 1/4 LSB)

Total Error At Step 0...001 (1/2 LSB)

(b) DAC

Total Error At Step 0...011
(1 1/4 LSB)
Questions?