SGS-THOMSON MICROELECTRONICS

APPLICATION NOTE

REAR MIRRORS MULTIPLEXING USING L9946

by L. Valsecchi & S. Vergani

The application of the L9946 device in a real-world automotive multiplex system is described. After a brief introduction to the multiplex concept, the hardware and software key points are discussed. It turns out that L9946 is very well suited for this kind of relatively complex applications.

THE MULTIPLEX CONCEPT

In this section a brief introduction to the basic multiplex (MUX) concepts is given. Generally speaking, a MUX is composed by a number of units connected through a serial bus. There is a set of meaningful serial messages and each unit can recognise a subset of messages relevant to it. Once a relevant message is received, the unit performs an action according to the information contained in the message. Usually an acknowledge technique is used, so that a bidirectional information flow between units can be established. It is possible to draw a rough distinction between MUX systems based on the communication strategy. In increasing complexity order, a MUX can be classified as follows.

1) MASTER-SLAVE:

One unit is qualified as master unit, and it is the only one that can autonomously start a transmission. The other units (slaves) can transmit only after the reception of a message out of a defined set.

2) QUASI MULTI-MASTER:

As before, one unit acts like a master, but some slave units can start an autonomous transmission to the master. This happens usually when a significant event has occurred (e.g. a key has been pressed).However, the slave units cannot communicate each other directly. The messages flow is under the total control of the master unit.

3) MULTI-MASTER:

Every units can commuicate each other, and there is no more a well defined master unit.In fact, the control, at a given time, is owned by the unit currently autonomously transmitting.

The format of the serial messages, as well as the characteristics of the physical interface of the bus line are defined by a series of rules called the PROTOCOL SPECIFICATION. These rules also define in details the behaviour of the transmitting and receiving units when a situation of bus contention (i.e. when two units try to access the bus simultaneously) occurs.

The ISO (International Organization for Standardization) has standarized, at various levels, three protocols called CAN, VAN, J1850. This means that documents exist as a reference to achieve the compatibility between two systems using the same protocol.

In fact, especially for slow speed data bus, custom protocols have been developed.

There are definite advantages in using a MUX system in the automotive field. First, the number of wires required to perform the same functions is dramatically reduced. For example,with the MUX approach, to connect a keyboard unit to other units such as window lift motor control, rear mirror control etc., only three (or four if a differential bus is used) wires are required, independently of the number of keys or motors used. This leads to a reduction of costs of the harness of the vehicle.

Flexibility is another feature common to well designed multiplex systems.

The multiplex architecture allows a high degree of freedom in the choice of the physical location of the units inside the vehicle. For example, as long as the serial bus line is provided, a control keyboard can be placed indifferently in the door or on the dashboard without changing the vehicle wiring.

Furthermore, if a certain computational power (i.e. a microcontroller) is located in the peripheral units, the functional behaviour of the whole system can be defined by software so that upgrading and modification can be accomplished without changing the hardware.

Also, a sophisticated diagnostic strategy can be implemented. Usually one or more units collect diagnostic information that can be read by a tester connected to the system when a car technical assistance is required. Such a tester generally includes a menu driven diagnosis procedure, lead-

ing to easier fault detection and thus to shorter repair time.

REAR MIRRORS MULTIPLEX SYSTEM

INTRODUCTION

The application described here is an example of how the L9946 can be used as a mirror controller in a MUX system.

To explain in all the details a MUX system design is beyond the scope of this application note. However, the key points in hardware and software design will be discussed in depth.

GENERAL DESCRIPTION

This MUX is composed by a keyboard unit, a left mirror unit and a right mirror unit.

The electronics is intended to be placed inside the external rear mirror case, and inside the physical keyboard. To achieve this, when possible, devices available in small SO package have been chosen. In this way, using surface mounting technique, very compact PCB layout can be obtained.

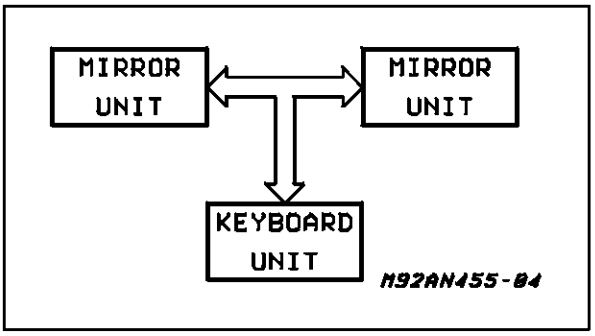
The three units are connected through a differential bus.

Including power supply line and ground return, only four connection wires are required, obtaining a substantial saving compared to the traditional solution, that requires eight wires.

The system block diagram is shown in fig. 1.

The functions implemented are:

Figure 1: System Block Diagram



- mirror plate movements
- open / fold
- wiper

The commands available are activated by 6 push buttons located in the keyboard unit.

An additional three-way selector allows to switch between the left or the right mirror. When this selector is in its central position, the only function available is a simultaneusly mirror open/fold movement.

MIRROR UNIT DESCRIPTION

The block diagram of the mirror unit is shown in fig. 2. The schematic diagram of the mirror unit is shown in fig. 3.
The only difference between the left and right mirror unit is the position of a jumper that configures the address of the unit.

Figure 2: Mirror Unit Block Diagram

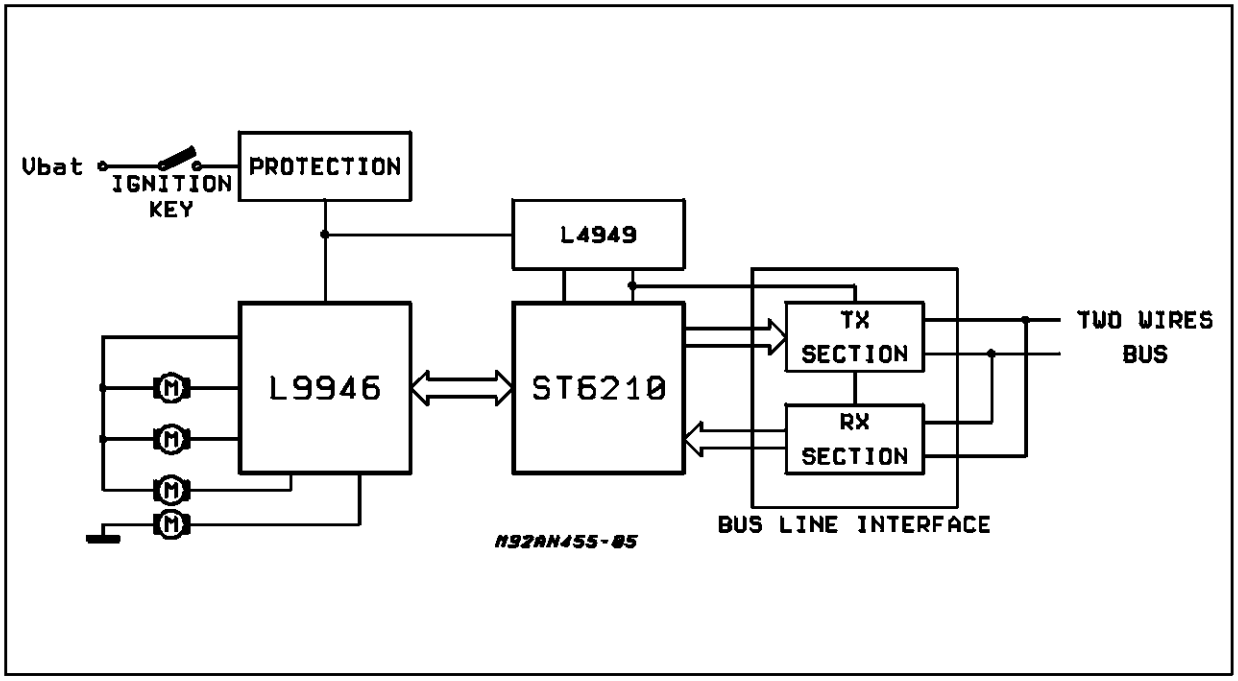
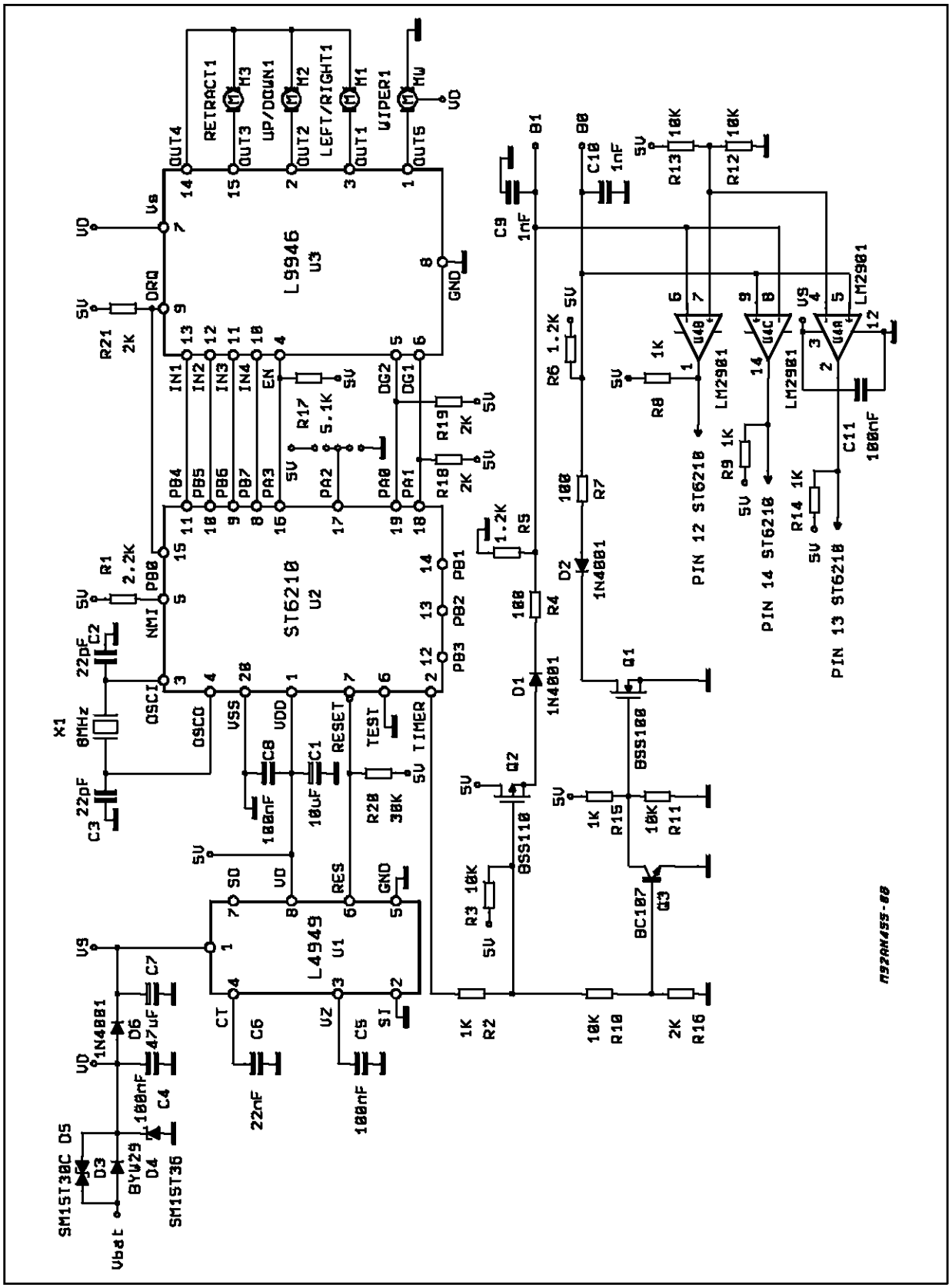


Figure 3: Mirror Unit Schematic Diagram.



In the following sections the main functional blocks are described.

Protections

The units must be protected against a number of possible anomalous voltages on the power supply line coming from the battery.

This anomalous conditions are:

1) Reverse battery

2) Load dump

3) Short negative spikes

Voltage Regulator (L4949)

This device provides the five volts necessary for the microcontroller and to the bus line interface. It also provides to the microcontroller the correct power on reset signal.

Microcontroller (ST6210)

The microcontroller (uC) choosen for this application is the ST6210 (1K8 EPRPOM, 64 bytes RAM).This is a 20 pin device, available in SO20 package. The ST6 family is intended for low-medium complexity applications.

The heaviest task for the uC in this, or similar, application is the protocol handling, i.e. the reception and the transmission of the serial messages on the bus.

Due to the low computational power and speed of this uC, the protocol was chosen to be relatively slow (3.3 kbits/sec) and the bit encoding was chosen in such a way that the decoding algorithm is tailored to optimize the hardware uC resource usage.

In this case the uC also drives the L9946 and protects it against overcurrent and/or overtemperature reading back the dignostic signals DG1 and DG2.

Mirror Actuator (L9946)

The L9946 in this application is used to drive the four mirror motors: two for the plate movements, one for the open/fold movement and one for the wiper motor. In this particular application the wiper can be driven by the high-side driver thanks to a mechanical solution built into the mirror that performs automatically the wiper alternative movement.

Bus Line Interface

This is the circuitry that realize the physical interface between the unit and the bus line. Since the functioning of the whole system relies on the correctness of the exchanged messages, the bus line interface must be designed very carefully. A complete discussion of the needed design criteria is far beyond the scope of this application note.

A list of desirable features is:

a) High noise rejection

b) Line faults (short to GND or VCC, wire cut) detection and real time recover.

c) High RF noise immunity

d) Low RF emission

The solution implemented here is a differential bus line driven by two complementary MOS devices.

The passive components around the MOS polarize and protect the devices against shorts, spikes and negative voltages applied to the bus lines.

Capacitances placed on the bus lines filter out RF noise, also reducing the bandwidth of the bus channel in order to avoid too sharp edges during bus transitions, i.e. RF emission and subsequently possible interferencies with other equipment (dashboard instrumentation, car radios etc.)

The three comparators in the RX section allow a full fault detection and recovery. This means that transmission and reception can continue also if one line is shorted to GND or V_{CC} or cut.

Figure 4: Keyboard Unit Block Diagram

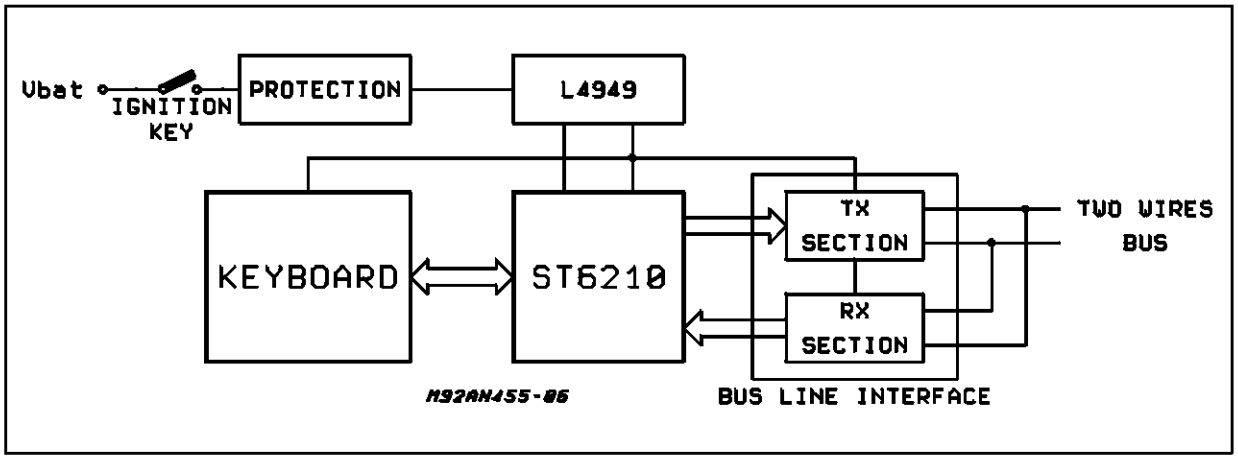
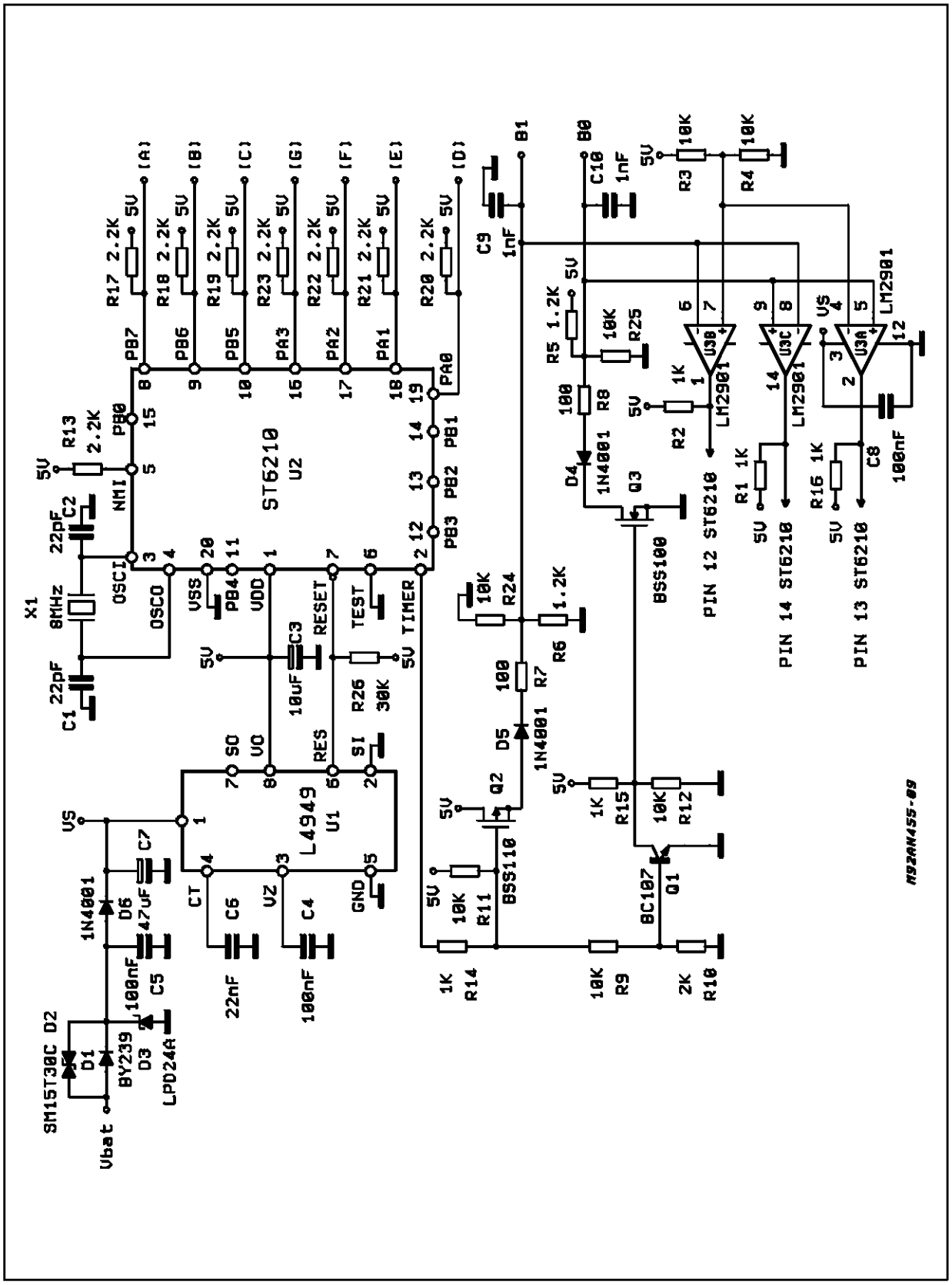


Figure 5: Keyboard Unit Schematic Diagram.



KEYBOARD UNIT DESCRIPTION

The block diagram and the schematic diagram of the keyboard unit are shown respectively in fig. 4 and fig. 5.

Many blocks in the architecture of this unit are very similar or identical to those used in the mirror unit.

This blocks are protections, voltage regulator and the bus line interface. The uC used is the same adopted in the mirror units.

A 4 X 2 matrix-organized keypad is connected to the unit. The schematic diagram of the physical keyboard is shown in fig. 6.

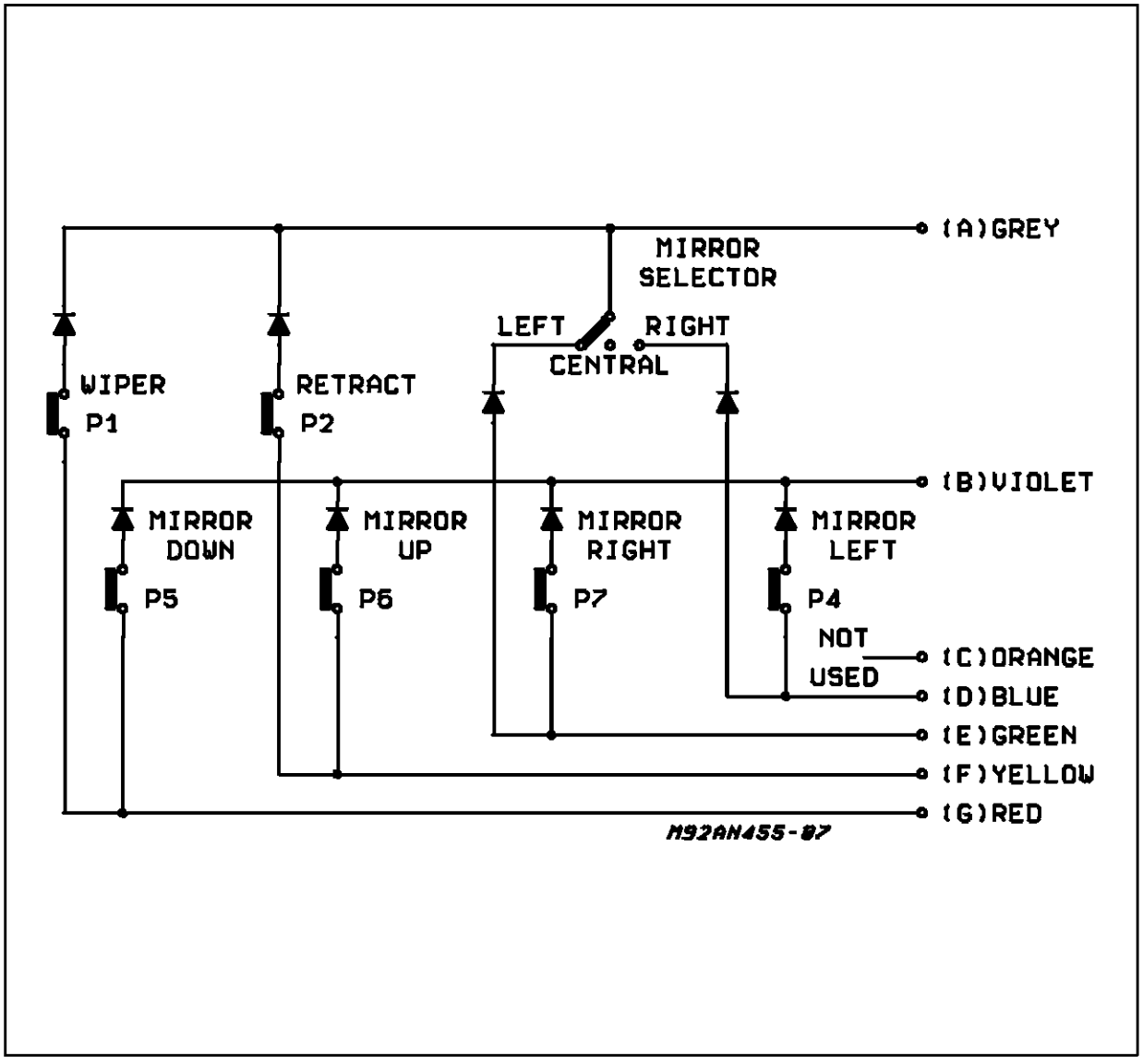
Since the physical keyboard rows and columns are directly connected to the uC pins, it must be placed very close to the electronics to avoid induced noise. A software debouncing strategy has been implemented so that a key transition is validated only if the new state has been stable for at least fifty milliseconds.

SOFTWARE DESCRIPTION

Introduction

Two different programs have been written, one for the mirror units and one for the keyboard unit. The software was developed using the ST6 hardware emulator, and the ST6 macroassembler and linker. The mirror unit and keyboard unit programs are respectively about 1340 and 1280 bytes long. The main difficult in this software development was to overcome the uC limitations (using some software tricks) without affecting the overall program's readability.

Figure 6: Keyboard Schematic Diagram.



Mirror Unit's Software Description

The software developed for this unit can be divided, at the functional level, into 3 main sections.

1) PROTOCOL HANDLER SECTION

This routines perform the serial bus message reception and transmission.

The reception procedure starts when a valid SOM (see BUS PROTOCOL DESCRIPTION paragraph) is detected. The reception ends successfully when the following conditions are obeyed:

a)Ten correct bits (i.e. with the right timing between two edges) are decoded.

b)The received checksum field matches the checksum calculated upon the preceding eight received bits.

If an error occurs, the reception is aborted and the unit starts to wait for a new valid SOM.

The transmission routine starts when the unit must send a message on the bus line. The ST6 timer is used to obtain the desired time between the edges.

2) MESSAGES DECODING AND ACTUATIONS

This section performs the message decoding and the actual driving of the L9946.

Once a correct message is received, the mirror unit compares the received address field with its own address. If they are equal, the data field of the message is decoded and the corresponding action or series of actions are undertaken.

The data fields recognized by the mirror units and their meaning are:

01110 : wiper on

10000 : wiper off

01010 : fold/open

00010 : up movement

00100 : down movement

00110 : left movement

01000 : right movement

10010 : stop motors

Immediately after the action has started, the mirror unit transmits an acknowledge message to the master unit (i.e. the keyboard unit). This message is the echo of the acknowledged reception.

3) L9946 PROTECTION

The L9946 DG1 and DG2 pins are connected to uC interrupts lines so that a fast switch off is executed when an overcurrent or an overtemperature occurs in the device.

Keyboard Unit Software Description

In the sotfware for this unit the protocol handler section is the same code used in the mirror units.

The keyboard units acts like the master of the system.

The matrix keyboard is scanned every five milliseconds. If no key status variation is detected, every fifty milliseconds a stop message is sent to the slave units as a polling. The slave units should answer to this messages. If this is not the case, the master unit knows that one or both slaves are disconnected or broken.

When a key is pressed, a debouncing procedure is started. If the pressure remains at least for fifty millisecond the corresponding command message is sent to the unit selected by the position of the three-way selector.

If this selector is in its central position, only the fold/open command is enabled, and the subsequent command is sent to both slave units.

The keyboard unit keep sending the command until the key is released. Then, the normal no-operation polling is executed.

BUS PROTOCOL DESCRIPTION

1. GENERAL

The information between the units is passed in messages transmitted serially on the bus connected to all the units.

When the bus is in the idle state, i.e. no message is transmitted, its state is called "passive state". It is driven in the "active state" by a transmitting unit at the start of a message for the "start of message" time.The state is passive for the first (most significant bit) information time, active for the next bit time and so on until the message is finished (terminated) in the passive state. The value of the bit is determined by the time elapsed between two consecutive transition of the bus state. This bit encoding is called VPWM (variable pulse width modulation).

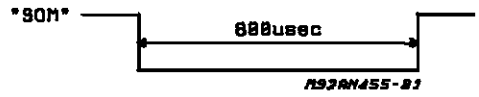
2. MESSAGE SYMBOL WAVEFORMS

The following sections show the nominal timing requirements of the VPWM message simbols generated by the software protocol handler as they appear on one wire of the bus. On the other bus wire the signal is inverted.

2.1. START OF MESSAGE

This symbol appears at the start of every message when a transmitter drives the bus in the active state to start the message.

2.2. DATA BIT

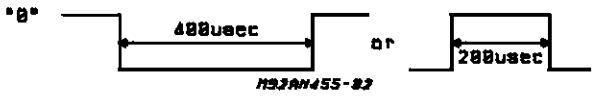


Each data bit is represented by the time between two consecutive transitions. These are both passive and active bit states that are used alternately.

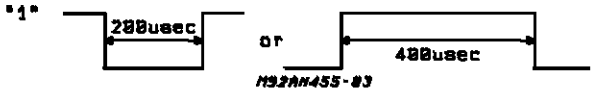
2.2. ”0” BIT

The two ”0” bit waveforms are:

2.3. ”1” BIT



The two ”1” bit waveforms are:



3. MESSAGE FORMAT

A message consists of a start of message (SOM) field, an address (ADR) field (3 bits), a data (DATA) field (5 bits) and a checksum (CHK) field (2 bits), for a total of 10 bits transmitted.

With the timing given in the above sections, the average transmission bit rate is 3.3 Kbps.

The SOM is the signal on wich every receiving unit starts the reception procedure.

Once a successful reception has been completed, the DATA field is decoded and the related action undertaken only if the ADR field matches with the wired address assigned to the receiving unit.

The CHK field is a way to detect some type of errors occurred during the DATA field bits transmission. During the reception procedure, a checksum value is calculated, and the reception is valid only if this value is equal to the contents of the received CHK field.

The algorithm used to calculate the checksum is the following.

a) Count the number of ”1” bits in the DATA and ADR fields.

b) Take the two less significant bits of this number.

c) Complement these bits.