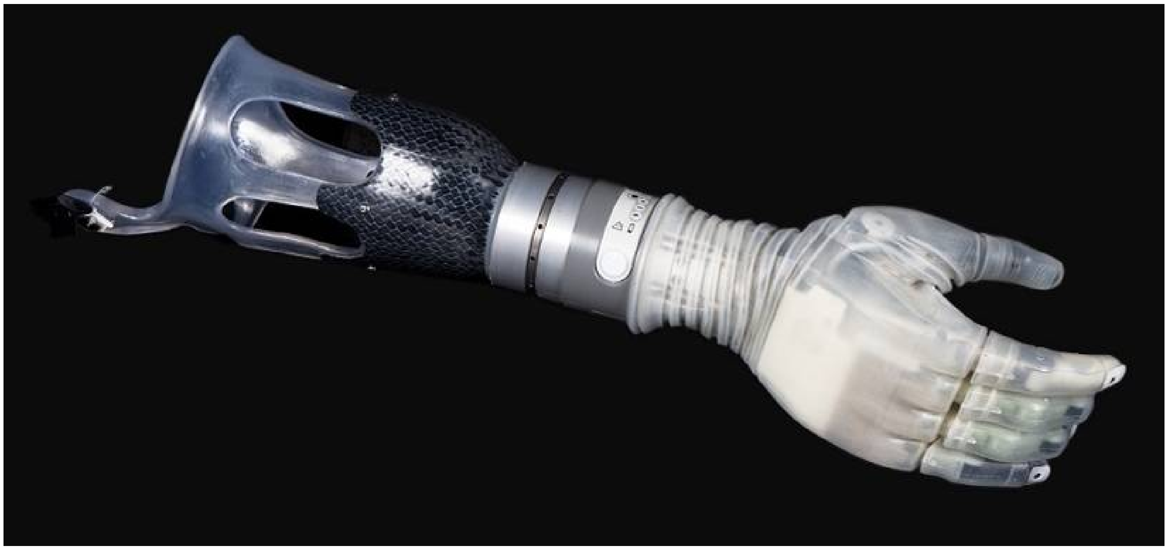
THE DEKA RC ARM

HARDWARE AND SOFTWARE INTERFACE SPECIFICATION

FOR THE HAPTIX PROGRAM



I. TABLE OF CONTENTS

II. INTRODUCTION

This document describes the general operation and hardware and software interface of the radial configuration (RC) DEKA Arm. This document was written to address the interface and operation of the DEKA Arm for the DARPA HAPTIX program.

III. OPERATIONAL OVERVIEW

The DEKA Arm is a prosthetic arm available in 3 different configurations to accommodate different levels of amputation. The radial configuration (RC) is for use by transradial (below elbow) amputees. The RC Arm has 6 powered degrees-of-freedom (DOF), including:

- Wrist rotator
- Wrist flexor with combined ulnar/radial deviation
- 2 DOF thumb
- Index finger
- Middle, ring, and pinky fingers that are driven by 1 motor and are mechanically linked to form a conforming grasp

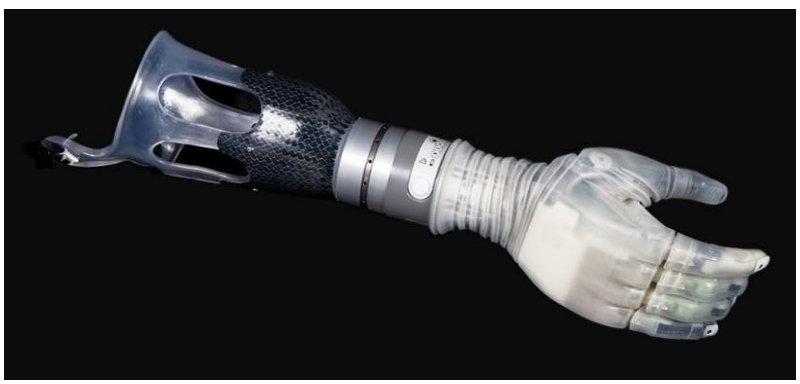


Figure 1: DEKA RC Arm

In typical use, an amputee controls the arm with industry-standard input devices such as EMG electrodes or with wireless sensors typically worn on the feet. These input devices control the velocity of a joint or function (such as hand open/close). A PC program called the Prosthetist Interface (PI) is used to map and configure an input device to a specific arm function.

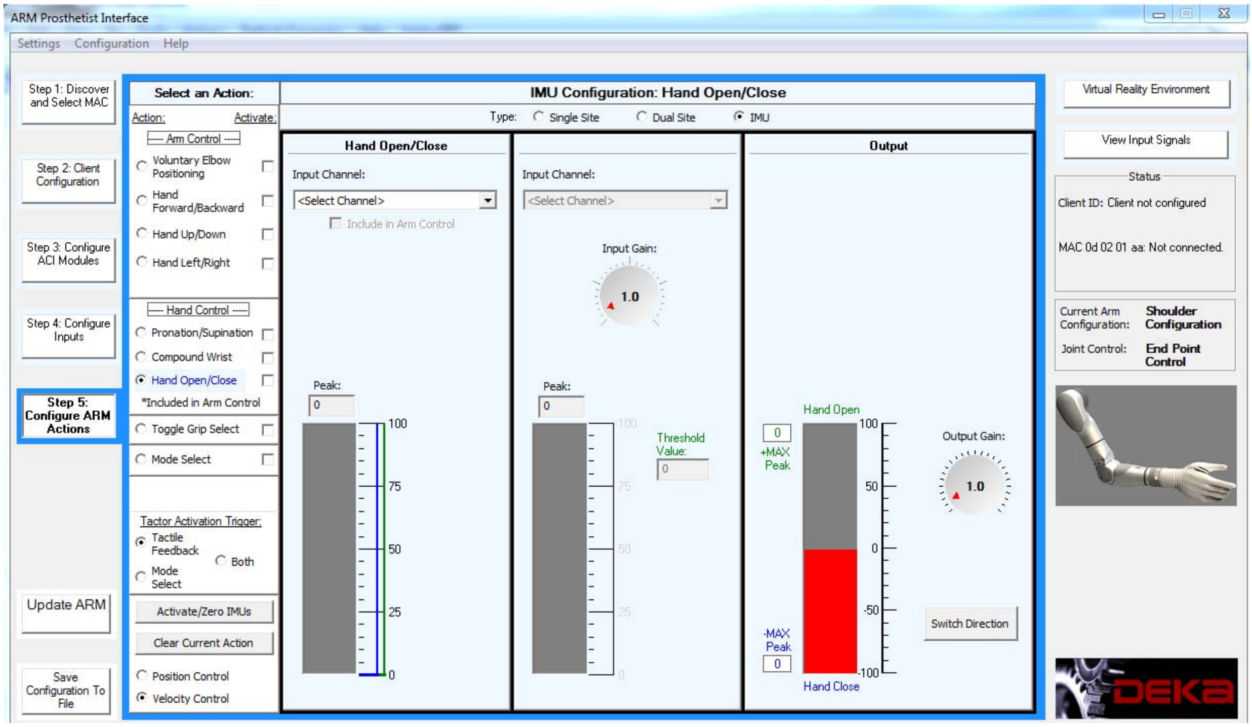


Figure 2: Prosthetist Interface

The arm has 6 pre-programmed grip patterns such as fine pinch, lateral pinch, and power. The user can cycle between grips when the hand is open, and then command the hand to open and close on the selected grip. For the HAPTIX program, the arm software has been upgraded to also allow individual fingers to be controlled in the hand. The hand control mode (grip select mode or direct finger control mode) can be configured through the PI software.

The arm is mounted to a socket which is fabricated by a prosthetist to fit an individual amputee. A socket interface piece, manufactured by DEKA, is fabricated into the socket by the prosthetist and allows the arm to mount to the socket. For more information on socket mounting, see Chapter 6 of the DEKA Arm Prosthetist Reference Guide.

IV. POWER

The arm is powered by a rechargeable battery with a nominal voltage of 14.8 V. The fully charged voltage is 16.4 V. The battery capacity is 5 Ah. Battery run time will vary depending on how the arm is used.

V. COMMUNICATION

The communication between various DEKA Arm modules and external devices is through a CAN bus interface. In the case of the HAPTIX program, an external device such as the neural interface processor sends messages on the CAN bus to emulate a DEKA Arm input device called the Arm Control Interface (ACI) module. Each ACI provides 4 inputs to the arm, and up to 4 ACIs can be connected to the DEKA Arm providing up to 16 input channels.

The software protocol is a command/response interface with the DEKA Arm acting as the bus master and the external device producing the response for between 1 and 4 ACIs. To control the RC DEKA Arm in direct finger control mode for the HAPTIX program, the external device emulates 4 ACIs.

The CAN interface is:

Baud rate:	1M baud
Refresh rate:	100 Hz (10 ms)
Message length:	8 bytes
Message type:	Standard

See the section IX information about the DEKA Arm CAN bus software protocol.

VI. ELECTRICAL INTERFACE

The main interface for power and communication to the DEKA RC arm is a cable that comes out of the socket interface. This cable is routed to a battery interface box that is typically mounted to the socket. A cable then runs out of the battery interface box to the battery holster which is typically worn at the waist (see Figure 3).

A second port on the battery interface box allows for interfacing devices to the arm's power and communication bus. In typical use, a device called the ACI (Arm Control Interface), which digitizes user inputs such as EMG electrodes, connects here to gather the input device signals to send to the arm. Each ACI can accept up to 4 input devices, and up to 4 ACIs can be connected to the arm at the same time, allowing for up to 16 input channels for control of the arm.

For the HAPTIX program, no ACI devices are used to control the arm. Instead, the neural interface processor from the HAPTIX teams connects to the arm and emulates multiple ACIs on the arm's communication bus.

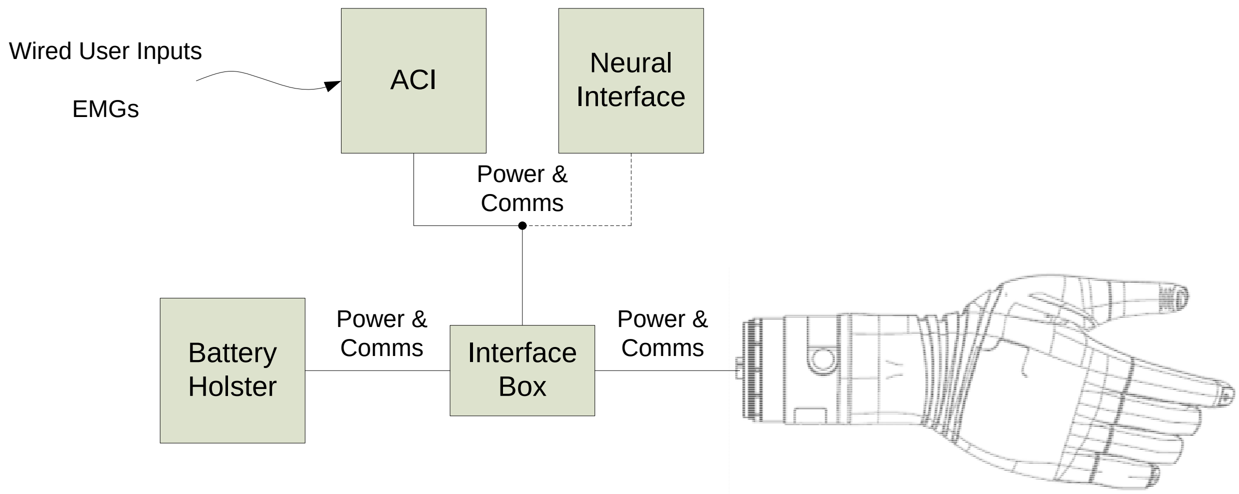


Figure 3: Arm Interconnect Overview

The power and communication connector at the battery interface box is Phoenix Contact part number 1530618.

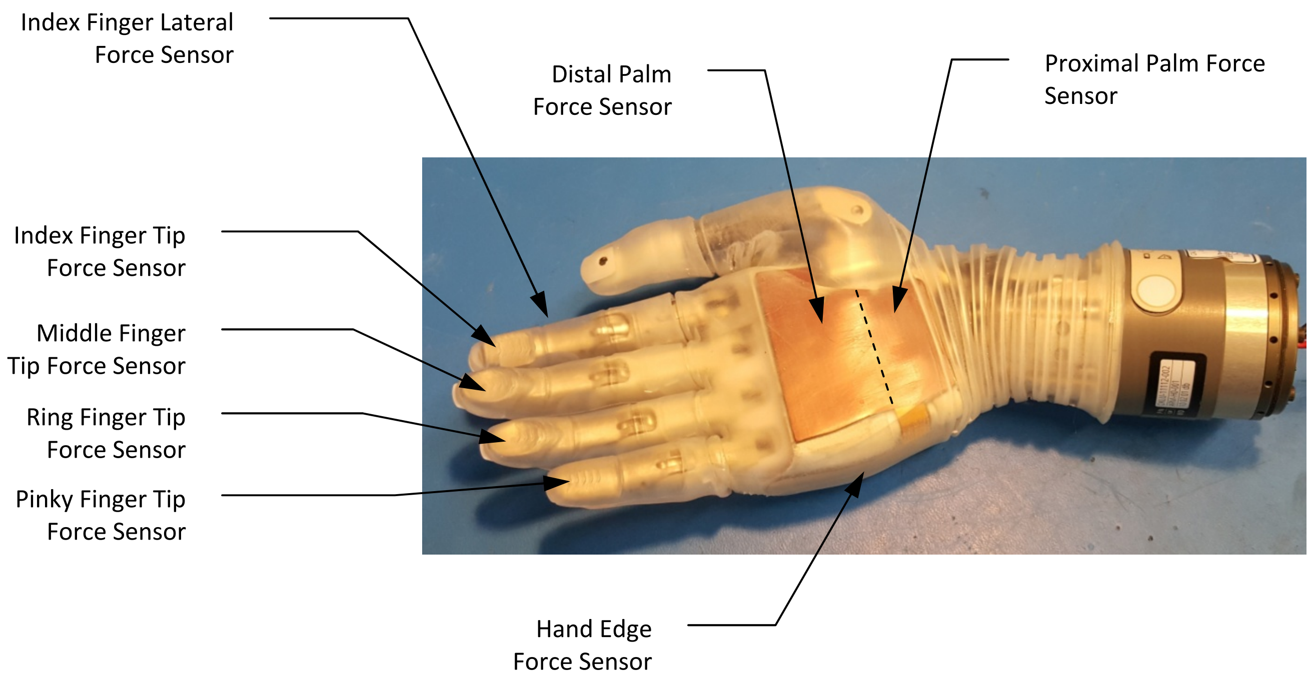
The mating connector is Phoenix Contact part number 1530304. See the pinout below. The wire color refers to the color as supplied with 1530304.

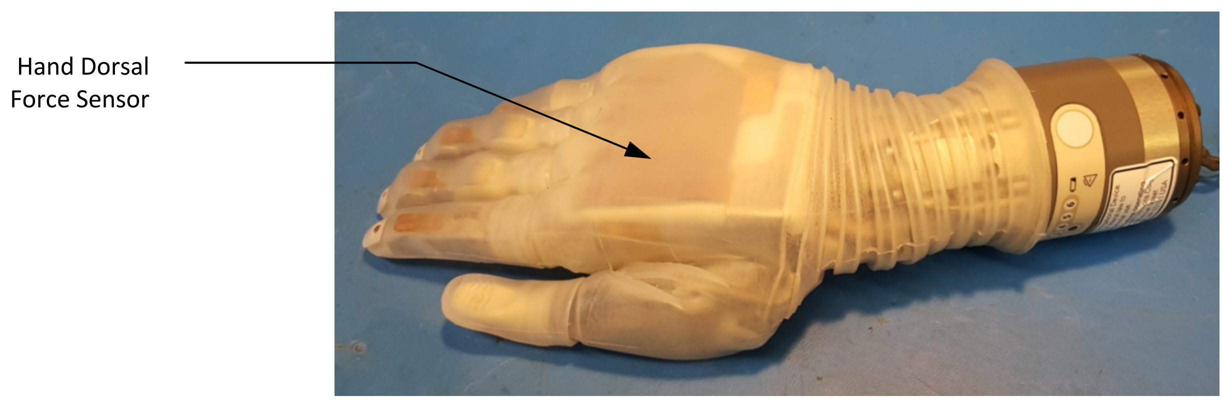
Pin Number	Wire Color	Function
1	Brown	Battery -
2	White	Battery +
3	Blue	CAN bus high
4	Black	CAN bus low

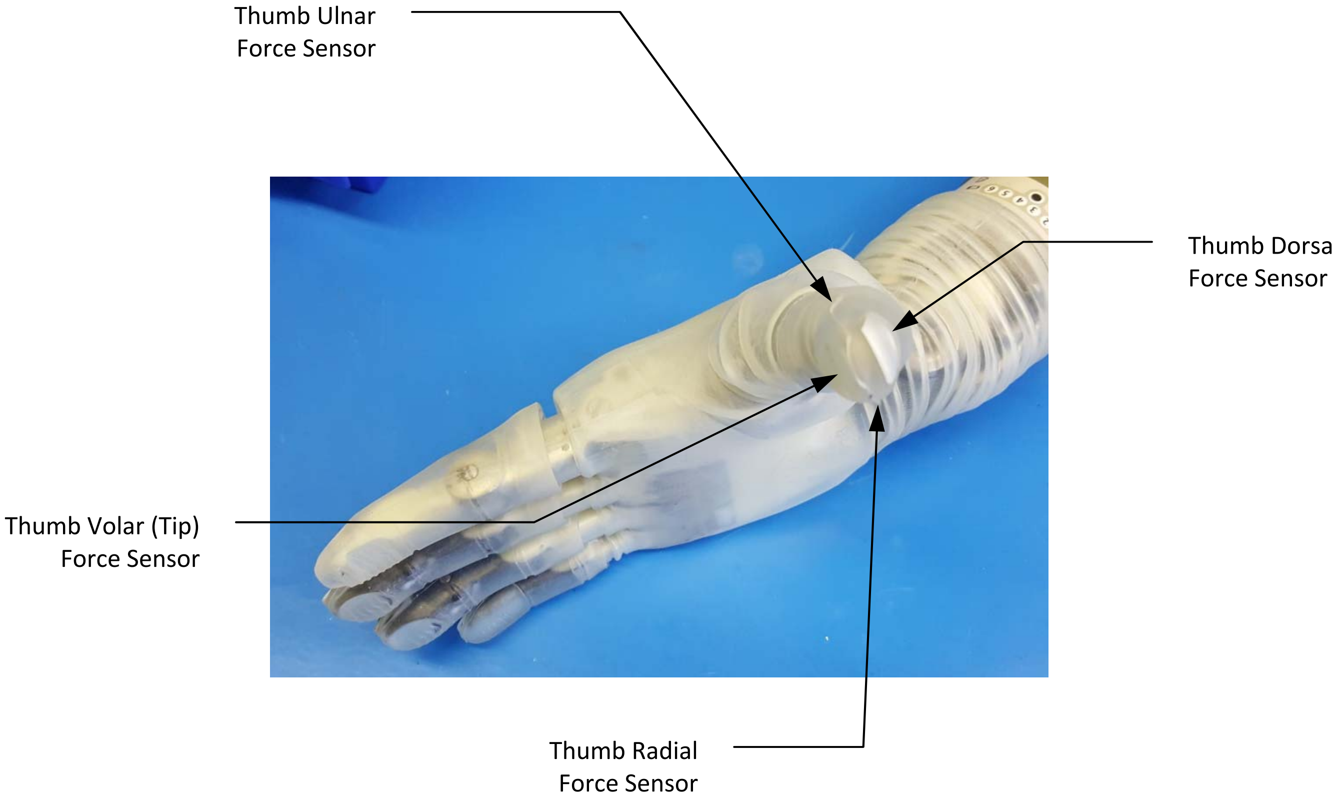
The DEKA arm contains the necessary CAN bus termination resistors required for proper operation – one at the hand and one at the battery holster for a termination resistance of approximately 60 Ohms. The HAPTIX team's neural interface does not need a CAN bus termination resistor.

VII. SENSORS

The DEKA Arm has many sensors, and this sensor information is available on the CAN bus interface. For example, each joint has an absolute position sensor. In addition, many force sensors have been added to the hand for the HAPTIX program. These new sensors are shown in the images below:


Index Finger Lateral Force Sensor
Distal Palm Force Sensor
Proximal Palm Force Sensor
Index Finger Tip Force Sensor
Middle Finger Tip Force Sensor
Ring Finger Tip Force Sensor
Pinky Finger Tip Force Sensor
Hand Edge Force Sensor


Hand Dorsal Force Sensor


Thumb Ulnar Force Sensor
Thumb Dorsal Force Sensor
Thumb Volar (Tip) Force Sensor
Thumb Radial Force Sensor

VIII. JOINT ANGLE VS. PHYSICAL POSITION

This section shows the physical position of each joint in the RC Arm vs. the reported angle on the CAN bus.

Index finger

Right hand is shown. Left hand behavior is the same (straight is 0°, flexed is 90°).

0°	90°
	

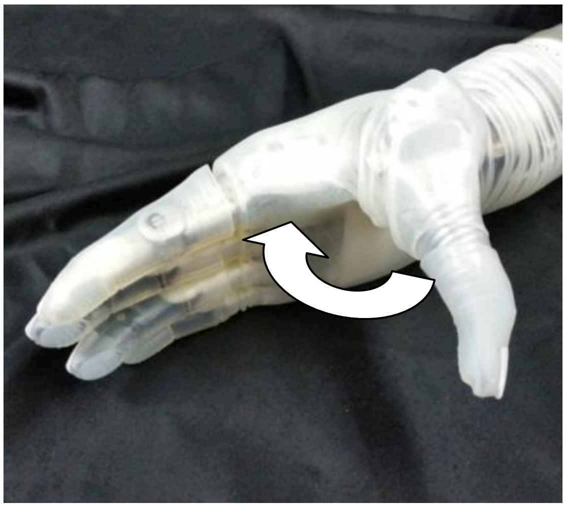
MRP fingers

Right hand is shown. Left hand behavior is the same (straight is 0°, flexed is 90°). NOTE: Since the MRP fingers form a conforming grasp, the reported position is an aggregate position of all 3 fingers.

0°	90°
	

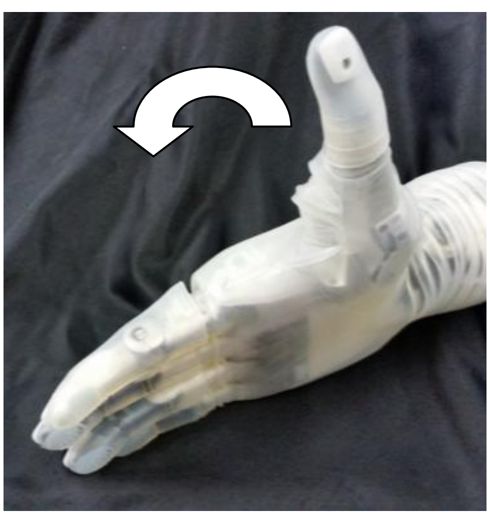
Thumb Yaw

NOTE: the thumb pitch is at 75° in the pictures below. Right hand is shown. Left hand behavior is the same (parallel to palm is 90°).

<table>
<tr><th>0°</th><th>90°</th></tr>
<tr><td></td><td></td></tr>
</table>

Thumb Pitch

NOTE: the thumb yaw is at 90° in the pictures below. Right hand is shown. Left hand behavior is the same (parallel to palm is 75°). Note that the thumb pitch can go beyond 75° (to 100°) when the thumb yaw is less than 90°, for example to align with the index finger for fine pinch.

<table>
<tr><th>0°</th><th>75°</th></tr>
<tr><td></td><td></td></tr>
</table>

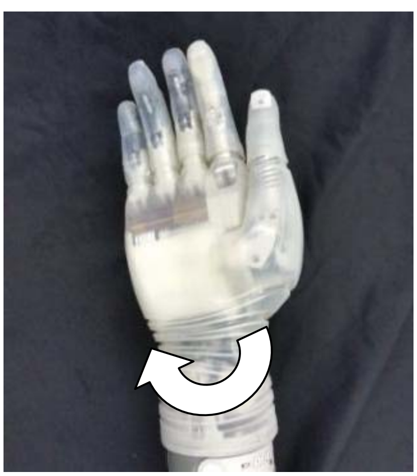
<u>Wrist Flexor-Deviator</u>

Right hand is shown. Left hand behavior is opposite (-55° is extended).

-55° flexed	+55° extended
	

<u>Wrist Rotator</u>

Right hand is shown. Left hand behavior is opposite (palm up is +120°).

-120° Palm up (supinate)	+175° Thumb down (pronate)
	

IX. SOFTWARE PROTOCOL

As bus master, the arm sends a sync message (message ID 0x080) every 10 ms. The external device responds to the sync message with 1 CAN message per ACI being emulated. The arm expects the external device to respond with the required response messages before the next sync message is sent.

In the command data fields in section B below, the expected range for commands is 0 to 1024 with 0 typically configured as “no movement”. These commands can be programmed to cause joint motion or change grips in the Arm, for example.

The mapping of arm inputs to arm actions is done through a DEKA Arm PC application called the Prosthetist Interface (PI).

The DEKA Arm also broadcasts sensor information on the bus. These messages are detailed in section C.

A. ACI MODULE EMULATION

The external device emulates 4 ACI modules on the CAN bus. Each ACI provides 4 analog inputs to the arm with 2 bytes of data for each of the 4 input channels (8 bytes total).

B. ACI EMULATION: COMMAND/RESPONSE SEQUENCE

1. The external device will receive the sync message (message ID 0x080) from the arm.

2. The external device then transmits message ID 0x210 (for ACI Module 1), with the command data fields configured as follows (max. command 1024):

ACI Module 1		
ACI Channel #	**Data Byte #**	**Description**
1	0	Thumb pitch up command – high byte
	1	Thumb pitch up command – low byte
2	2	Thumb pitch down command – high byte
	3	Thumb pitch down command – low byte
3	4	Thumb yaw right command – high byte
	5	Thumb yaw right command – low byte
4	6	Thumb yaw left command – high byte
	7	Thumb yaw left command – low byte

3. The external device then transmits message ID 0x211 (for ACI Module 2), with the command data fields configured as follows (max. command 1024):

ACI Module 2		
ACI Channel #	**Data Byte #**	**Description**
1	0	Index flex command – high byte
	1	Index flex command – low byte
2	2	Index extend command – high byte
	3	Index extend command – low byte
3	4	MRP flex command – high byte
	5	MRP flex command – low byte
4	6	MRP extend command – high byte
	7	MRP extend command – low byte

4. The external device then transmits message ID 0x212 (for ACI Module 3), with the command data fields configured as follows (max. command 1024):

ACI Module 3		
ACI Channel #	**Data Byte #**	**Description**
1	0	Wrist rotator pronate command– high byte
	1	Wrist rotator pronate command– low byte
2	2	Wrist rotator supinate command– high byte
	3	Wrist rotator supinate command– low byte
3	4	Wrist flexor/deviator extension command– high byte
	5	Wrist flexor/deviator extension command– low byte
4	6	Wrist flexor/deviator flexion command– high byte
	7	Wrist flexor/deviator flexion command– low byte

5. The external device then transmits message ID 0x213 (for ACI Module 4), with the command data fields configured as follows (max. command 1024):

ACI Module 4		
ACI Channel #	**Data Byte #**	**Description**
1	0	Mode select command – high byte
	1	Mode select command – low byte
2	2	Not used. Send 0x00.
	3	Not used. Send 0x00.
3	4	Not used. Send 0x00.
	5	Not used. Send 0x00.
4	6	Not used. Send 0x00.
	7	Not used. Send 0x00.

C. SENSOR DATA

The DEKA Arm broadcasts sensor information on the CAN bus using the messages shown in this section.

1. Wrist and Index Finger Positions – Message ID 0x4AA:

   Data[0] = Wrist rotator position (degrees) X 2^6 – high byte

   Data[1] = Wrist rotator position (degrees) X 2^6 – low byte

   Data[2] = Wrist flexor/deviator position (degrees) X 2^6 – high byte

   Data[3] = Wrist flexor/deviator position (degrees) X 2^6 – low byte

   Data[4] = Index finger position (degrees) X 2^6 – high byte

   Data[5] = Index finger position (degrees) X 2^6 – low byte

   Data[6] = MRP fingers aggregate position (degrees) X 2^6 – high byte

   Data[7] = MRP fingers aggregate position (degrees) X 2^6 – low byte

2. Thumb position sensors – Message ID 0x4BF:

   Data[0] = Thumb pitch raw position (degrees) X 2^6 – high byte

   Data[1] = Thumb pitch raw position (degrees) X 2^6 – low byte

   Data[2] = Thumb yaw raw position (degrees) X 2^6 – high byte

   Data[3] = Thumb yaw raw position (degrees) X 2^6 – low byte

   Data[4] = Thumb yaw position (degrees) X 2^6 – high byte

   Data[5] = Thumb yaw position (degrees) X 2^6 – low byte

   Data[6] = Thumb pitch position (degrees) X 2^6 – high byte

   Data[7] = Thumb pitch position (degrees) X 2^6 – low byte

3. HAPTIX force sensors #1 – Message ID 0x241:

   Data[0] = Index lateral force sensor (N x 10)

   Data[1] = Index tip force sensor (N x 10)

Data[2] = Middle tip force sensor (N x 10)

Data[3] = Ring tip force sensor (N x 10)

Data[4] = Pinky tip force sensor (N x 10)

4. HAPTIX force sensors #2 – Message ID 0x341:

   Data[0] = Distal palm force sensor (N x 10)

   Data[1] = Proximal palm force sensor (N x 10)

   Data[2] = Hand edge force sensor (N x 10)

   Data[3] = Hand dorsal force sensor (N x 10)

5. HAPTIX force sensors #3 – Message ID 0x4C2:

   Data[0] = Thumb ulnar force sensor (N x 10)

   Data[1] = Thumb radial force sensor (N x 10)

   Data[2] = Thumb volar (tip) force sensor (N x 10)

   Data[3] = Thumb dorsal force sensor (N x 10)